Y, CALIFORNIA 93943-5002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

M33545

MODELLING ARMS TRANSFERS: AN APPLICATTION OF KSIM CROSS IMPACT ANALYSIS

by

Michael R. Mara

December 1987

Thesis Co-Advisors: F. Russell Richards
Douglas E. Neil

Approved for public release; distribution is unlimited

T239089

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) MODELLING ARMS TRANSFERS: AN APPLICATION OF KSIM CROSS IMPACT ANALYSIS

12 PERSONAL AUTHOR(S) MARA, Michael R.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	December 1987	76

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	KSIM, Cross Impact, Cross Impact Analysis, Arms Transfers

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The arms transfer process is modeled using KSIM, an enhanced version of cross impact analysis. The cross impact technique of forecasting is developed from its origin in the Delphi procedure through several of the key improvements to the original method. The KSIM approach to cross impact is discussed in detail, and is tailored to model arms transfers. A method of obtaining initial parameter estimates from an existing data base is described. The model is run and several proposed arms transfer policies are simulated. The implications of these policies are then analyzed.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Prof. F. Russell Richards	408-372-3439	55Rh

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

Modelling Arms Transfers: An Application of KSIM Cross Impact Analysis

by

Michael R. Mara
Lieutenant, United States Navy
B.S., United States Naval Academy, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

ABSTRACT

The arms transfer process is modeled using KSIM, an enhanced version of cross impact analysis. The cross impact technique of forecasting is developed from its origin in the Delphi procedure through several of the key improvements to the original method. The KSIM approach to cross impact is discussed in detail, and is tailored to model arms transfers. A method of obtaining initial parameter estimates from an existing data base is described. The model is run and several proposed arms transfer policies are simulated. The implications of these policies are then analyzed.

TABLE OF CONTENTS

I. INTRODUCTION 8

II. CROSS IMPACT ANALYSIS REVIEW 11
- A. INTRODUCTION 11
- B. DELPHI 12
- C. CROSS-IMPACT ANALYSIS 14
- D. CROSS IMPACT IMPROVEMENTS 15
  - 1. Enzer 15
  - 2. Turoff 17
  - 3. Helmer 18
  - 4. Bloom 19
  - 5. SMIC-74 20
  - 6. KSIM 22
- E. EVALUATION 25

III. DEVELOPMENT OF AN ARMS TRANSFER MODEL 26
- A. INTRODUCTION 26
- B. KSIM PHILOSOPHY 26
- C. KSIM MODEL OF ARMS TRANSFERS 27
- D. MODEL PROPERTIES 29
- E. DETERMINATION OF Q(T) 31
  - 1. Event Parameter Equations 33
- F. EVENT PARAMETER ESTIMATION 35
  - 1. Event Data Set 36
  - 2. Data Set Coding 37
  - 3. Parameter Estimation Technique 40
- G. KSIM ALGORITHM 42

IV. RESULTS AND CONCLUSIONS 44
- A. MODEL RESULTS 44

1. Input Parameters 44
2. Promoting the Transfer of Defensive Weapons 54
3. Extended Congressional Debate 54
B. CONCLUSIONS 56

APPENDIX: KSIM ARMS TRANSFER SIMULATION PROGRAM LISTING 59

LIST OF REFERENCES 73

INITIAL DISTRIBUTION LIST 75

LIST OF TABLES

1. EVENT DATA RECORD 39
2. DATA FOR ESTIMATION OF EVENT PARAMETERS 41
3. SAUDI ARABIA ARMS TRANSFER ACTIVITY (1985 - 1986) 42
4. INPUT PARAMETERS FOR KSIM SIMULATION 44
5. DERIVATIVE CROSS IMPACT MATRIX 48
6. CONSTANT CROSS IMPACT MATRIX 49

LIST OF FIGURES

3.1 Logistics Curve PDF 32
3.2 Logistics Curve CDF 34
3.3 Arms Transfer Process 38
4.1 KSIM Simulation With No Cross Impacts (0 - 0.5) 46
4.2 KSIM Simulation With No Cross Impacts (0 - 0.1) 47
4.3 Baseline Run of KSIM Model (Full Plot) 50
4.4 Effect of a Large Negative Impact of Order on Reject/Refuse 52
4.5 Effect of a Large Positive Impact on Deliver 53
4.6 Results of Increased Effort in Propose/Request 55
4.7 Results of Extended Congressional Debate 57

I. INTRODUCTION

"The global trade in conventional armaments has become a burning political issue in recent years, owing primarily to the dramatic increase in the volume of the international market since the early 1970's." [Ref. 1: p. 1] The new significance in world arms sales is the result of three factors. [Ref. 2: pp. 275-276] The first is the sheer volume of arms being traded and the increasing quality of these weapons. It used to be the norm that countries would sell only obsolete, unneeded or second rate equipment. Now nations are exporting state of the art weapon systems with unparalleled accuracy and destructive capacity. The second factor is the decline of effectiveness in alternative methods for the superpowers to influence other countries such as diplomacy, alliances and the threat of direct intervention. In today's political climate, the superpowers are much less likely to intervene in local hostilities with their own armed forces; they are much more likely to attempt to support their interests with the supply of weapons. Finally, the past decade has seen a dramatic increase in the demand for weapons by Third World nations. As new Third World nations have been created, and others have continued to develop, they have sought to acquire the weapons of the industrialized world. Given the important role of arms transfers in global politics, then, some measure of control over this activity would be extremely desireable. Andrew Pierre observes, "Arms transfers ... should be managed so as to prevent or contain conflict and enhance the forces of moderation and stability." [Ref. 2: p. 7]

One avenue of arms control might be the voluntary restraint of the recipient nations. To see if this approach is plausible, the motivation of the recipient for procuring weapons must be examined. There are a variety of reasons why Third World countries purchase weapon systems; the need for defense from external aggression, an indication of regime legitimacy and a desire for world prestige. However it can be easily argued that regime survival is the primary reason for buying arms. Thus, it is the immediate needs of the Third World regimes that dominates their perceived needs for arms imports. The regimes are generally not concerned with the global problem of escalating arms levels or of the balance of power; they simply want to remain in power and maintain their capability to suppress internal insurgencies. In an environment

such as this, it is inconceivable that recipient nations would agree to any form of voluntary arms import restraints.

What then of supplier restraints? Might the major arms suppliers agree to exert a collective measure of control over arms transfers? The United States and the Soviet Union account for the majority of world arms transfers and it is fair to say that the primary reason these nations transfer weapon systems is for political and ideological considerations. However, second tier suppliers such as France and Brazil, are heavily motivated by economic factors. "France has often been cited as the best example of a country whose arms trade policies are determined more by commercial than political motivations. Long before the 1973 energy crisis, the French (and to a lesser extent the British) sought the most lucrative available markets and were largely uninhibited by political restraints." [Ref. 1: p. 69] Similarly, Brazil exports 95% of its arms production [Ref. 3: p. 87]. They make no distinction regarding prospective customers - in fact they currently sell arms to both Iran and Iraq. When the dichotomy between US and Soviet ideologies is considered, it is unlikely that these two superpowers will be able to agree on bilateral arms transfer constraints. Furthermore, considering the economic motivation of second tier nations, it is just as unlikely that these countries would find satisfactory multilateral constraints. Thus, unilateral constraint by the United States appears to be the only likely avenue of exerting any measure of control over the transfer of weapons systems.

What are some examples of these controls that the US may attempt to implement? Gompart and Vershbow observe that, "It must be recognized that some types of arms, by their technical nature, are generally stabilizing, especially those that can be more effectively employed for defensive than for offensive purposes." [Ref. 1: p. 13] It may be in the best interest of the US to promote the transfer of these weapons. But what actions might the US take to accomplish this and how effective would these measures be? Another example of a form of control over transfer of arms proposed by Hessing, et al, is to make the transfer process more viscous by extending the time for debate in Congress [Ref. 1: p. 100]. What would be the result of such a policy? Would it have any effect on the likelihood of transfers to the recipient? To answer questions like these, the foreign policy analyst would need some sort of model of the arms transfer process so that he could test the various policies and see what the likely effect would be.

Laurance and Mullen note that, "most of the scholars and analysts conducting research on arms transfer issues are generalists and political scientists drawn to the issues by concern for policy and the policymaking process." [Ref. 3: p. 89] They felt that there is a need for specialists from more diverse fields of study to be brought into the investigation of arms transfers. The purpose of this study is to bring Operations Research techniques to bear and develop a model of the arms transfer process. The model will be structured so that it can be used as an arms transfer policy tool by analysts with very limited mathematical background. The technique chosen to model arms transfers is cross impact analysis. The next chapter will introduce the concept of cross impact analysis and KSIM, an enhancement to the original technique. Chapter 3 will tailor KSIM to model the arms transfer process and will be followed by a chapter discussing the results of the simulation.

II. CROSS IMPACT ANALYSIS REVIEW

A. INTRODUCTION

Futures research is a discipline that explores different techniques of forecasting. "The basic mission of futures research is to broaden our time horizons and enable us not only to anticipate long term change per se, but also to see how, by controlling such changes, we can increase the range of our alternatives and select alternatives likely to produce a better society in both the near and longer time periods." [Ref. 4: p. 30] Futures researchers tend to make a distinction between prediction and forecasting. In their view, prediction is an attempt to specifically identify the future as a series of events. If one takes this view with the future regarded as inevitable, and nothing could be done to alter this series of events, then the study and analysis of the future would be pointless. Forecasting, however, seeks to identify the alternative futures, estimate the relative likelihood of these alternatives and investigate what controls exist to change the likelihoods. In order for a forecast to be believable, it must be supported by solid analytical methods. The development of analytical tools for the purpose of obtaining credible forecasts is the cornerstone of futures research.

Most forecasting techniques, such as moving averages, regression, and various smoothing routines, require substantial data to implement. However, some systems are so unique that no hard trend data exists. Furthermore, because the future of some systems is so evolutionary, simple extrapolation of trend and growth curves is an unsatisfactory method for developing a plausible forecast. Similarly, although classical probability and statistics may deal with time, it is always in terms of the past or present rather than the future. For this reason standard statistical techniques have been found to be lacking in their ability to analyze the future. Thus, in order for futures research to be effective, it must employ techniques that take advantage of judgemental data, since most of the insight necessary for the anticipation of unique events is simply not available through any other means. As Helmer points out, "it must be recognized that futures analysis, like operations analysis, of which it should properly be considered a part, is inevitably conducted in a domain of what might be called 'soft data' and 'soft laws'. This means that dependance on intuitive judgement is not just a temporary expedient but in fact a mandatory requirement." [Ref. 5: p. 18]

Futures research must also involve inputs from fields related to the area of interest, although not directly part of that area. This is because changes in one area of society may have a "spill over" effect and , conversely, changes in other aspects of society may affect the area of study. Finally, the research must be systematic, as a strict methodological approach is often the only way to sort through the many complex issues that face the futures researcher. The concept of using an expert, or a committee of experts, in some sort of structured analysis soon emerges as one the most useful methods available in the field of futures research.

The purpose of this chapter is to show that cross impact analysis is a useful method of incorporating expert opinion into an analytical model. The chapter will present the Delphi technique and demonstrate how the original cross impact model was developed to rectify perceived shortcomings in Delphi. A review of cross impact enhancements will follow. This review is not comprehensive; it only presents what this researcher feels are the most widely used enhancements. Nor does the review of the various techniques go into enough depth to stand alone; readers with further interest in any of these developments are refered to the original papers.

B. DELPHI

The Rand Corporation was prominent among the early pioneers in futures research. They observed how well expert opinion fit into the requirements described above, but believed that a group of experts was eminently more capable of providing an accurate forecast of the future than a single expert. "The basic notion was that the group would interact to compensate for the biases of individual members, and that the knowledge of one member of the group may compensate for another's ignorance." [Ref. 6: p. 18] However, assembling a group of experts in one room was not always conducive to achieving the goal of the group; namely arriving at the best possible forecast. The reason was that group dynamics often became the prime motivater of the panel, rather than a common view of the objective. Some of the specific problems associated with group dynamics are:

1. A group may exert significant pressure on its members to conform with the group view, even though the group view may be wrong.
2. A strong vocal minority may overwhelm the majority by the sheer volume of their arguments while the merit of their arguments, taken individually, may be minimal.

3. In groups where no official leader has been appointed, any one individual may have an undue impact on the outcome of the committee decision based on the strength of his personality, not the merit of his arguments.
4. Groups are often more interested in reaching agreement on a forecast rather than finding a forecast that may be more accurate, yet offend some committee members.
5. The group, as a whole will probably share a common bias based on its culture, education, and field of expertise.

The Delphi procedure of forecasting was developed by the Rand Corporation as a way to eliminate most of the disadvantages of group dynamics while capitalizing on the collective knowledge inherent in a group of experts. The entire Delphi procedure was conducted via a series of questionnaires; the panel was never required to be assembled in a common location for the procedure to work. The identity of group members was not made known to individual members which avoided a specific opinion being linked to a specific panelist. The questionnaires were structured in order to provide feedback to the group at each step of the procedure. At the conclusion of the Delphi, the director had a statistical summary which included the opinions of the entire group. A brief outline of the "classic" Delphi as described by Martino [Ref. 6] follows:

Step 1. Panelists are asked to provide a forecast of the future in the subject area of interest. This step is completely unstructured; some panelists may provide a narrative scenario while others may provide a list of events and associated dates. The director must take these inputs and consolidate them into a final list of key events that are as specific as possible.

Step 2. This list of events is returned to the group who are asked to estimate the date by which the event will have occured. In addition, the group members must state a reason why they chose that date as their estimate. The director then prepares a statistical summary of the groups opinions (median and inter-quartile range for each event) and summarizes the arguments supporting the date estimates.

Step 3. The Delphi panel is now presented with a list of events, the statistical summary, and reasons for date estimates. Individuals are asked to review the arguments and make a new estimate for the date by which an event will have occurred. If the new estimate falls outside the inter-quartile range, the panelist is asked to justify his view and comment on opposing arguments. The director now computes new medians and inter-quartile ranges for the updated estimates and again summarizes the arguments.

Step 4. The updated medians and inter-quartile ranges are presented to the panel for one more evaluation. As in **Step 3**, if estimates fall outside the inter-quartile range, panelists must justify their extreme views. The director can now compute a final median and inter-quartile range, and he has a set of arguments germane to those events for which no date was settled on. The forecast, then, consists of the list of events and their median and inter-quartile dates.

Often, there were events included in a Delphi analysis that were inter-related. The way that the procedure was structured required that estimates for the occurrence of one of these inter-related events be given without consideration for the dates of any other events. If the occurrence of one event was predicated upon the prior occurrence of another event, this proved to be a serious problem in providing accurate estimates. It was for this specific reason that cross impact analysis procedure was created. "The genesis of cross impact was the problem that Delphi panelists were sometimes asked to make forecasts for individual events, when other events in the same Delphi could affect these events. Thus, it was recognized that there was a need to allow for these cross impacts from one event to another." [Ref. 7: p. 61]

C. CROSS-IMPACT ANALYSIS

Cross impact analysis was first developed by T. J. Gordon and H. Hayward in 1968. Their goal was to overcome the difficulty of event interrelationships by estimating and explicitly accounting for them. They reasoned, "Most developments are in some way connected with other events and developments. It is hard to imagine an event without a predecessor that made it more or less likely or influenced its form - or one which, after occurring, left no mark. This interrelationship between events is called 'cross impact'." [Ref. 8: pp. 100-101] Events connected by cross impacts had two modes; event A could either enhance or inhibit the occurrence of event B. These event linkages were then assigned a strength such that a high strength indicated the occurrence of event A exerted a large influence on the probability of occurrence of event B. Using this data, a cross impact matrix, S_{ij}, was developed showing how the occurrence of every event j impacted on the probability of occurrence of every other event i. Once an initial probability of occurrence, P(i), for each event was estimated, the model was ready to be tested.

The cross impact model developed by Gordon and Hayward was stochastic in nature. They felt that after a large number of runs (on the order of 1000), steady state

probabilities for each of the events would be reached. The basic procedure is described here.

Step 1. Chose an event, E_i, at random from the list of events. Using a uniform random number generator (0,1), determine whether or not E_i occurred based on its initial probability of occurrence P(i).

Step 2. If E_i did not occur, discard it from the list and choose another event. Continue this process until an event is found to have occured.

Step 3. Modify the probabilities of occurrence of all remaining events according to the equation:

$$P(j|i) = P(j) + P(j)(1 - P(j))S_{ij}$$

Step 4. Return to **Step 1** as many times as necessary until no events remain.

This four step procedure constituted a single run and was repeated 1000 times. The final probability of occurrence for each event was estimated by noting how many times an event occured in the 1000 runs and dividing by 1000.

Gordon and Hayward then tabulated the results by event, initial and final probabilities, and a ranking according to initial probability and change in probability. From these tabulations, they drew conclusions about which events were most susceptible to change and which were most resistant to change. "The ranking by probability shift is, in essence, a list of the items most affected by the suspected interaction. In other words, the item which had the highest probability shift could be expected to be the one most influenced by external events depicted by the remainder of the list." [Ref. 8: p. 108] One of the real values of cross impact, though, was found to be the ability to test the effect of various policy decisions on the final occurrence probability of the events. The evaluation of policy decisions could be implemented by varying the probability of one or more events, replaying the matrix, and comparing these results to the original results.

D. CROSS IMPACT IMPROVEMENTS

1. Enzer

While cross impact had a great deal of intuitive appeal, there were significant drawbacks to the original approach. Gordon and Hayward noted, "We believe that this work is only indicative of a methodology of cross impacts. If possible, its current

shortcomings should be corrected in future work." [Ref. 8: p. 115] Selwyn Enzer recognized some of these shortcomings and sought to correct them. Enzer observed that since the relationship between an event's initial probability and its final probability was based on a quadratic function, it was impossible for an inhibiting event to significantly change a high probability and an enhancing event to significantly change a low probability event. Moreover, the effect of opposite cross impacts were not symmetric. For these reasons, Enzer abandoned the quadratic manipulation suggested by Gordon and Hayward and developed a cross impact method based on the likelihood ratio. [Ref. 4] An overview of this model follows.

The odds of an event occuring, O(i), are computed from the probability of occurrence, P(i), as follows:

$$O(i) = P(i)/[1 - P(i)]$$

Similarly, probability can be computed from the odds by

$$P(i) = O(i)/[1 + O(i)]$$

If event j occurs and changes the odds of event i, then the likelihood ratio, R_{ij} is related to the odds of i given j occured by the following:

$$O(i|j) = R_{ij}O(i)$$

This implies that

$$P(i|j) = O(i|j)/[1 + O(i|j)]$$

Using these relationships, then, the modified probability of the occurrence of i, given j has occured, is

$$P(i|j) = R_{ij}P(i)/[1 + (R_{ij} - 1)P(i)] \qquad \text{(eqn 2.1)}$$

Enzer used the same basic Monte Carlo approach of Gordon and Hayward, but modified the probabilities after cross impact by equation 2.1 Through the use of the likelihood multiplier, the magnitude of change in initial probability is the same for reciprocal likelihood ratios and "the domain of change that is permitted by this technique extends from 0 to infinity, so that impacting events can have the effect of totally eliminating the possiblity of occurrence of a subsequent event, or in fact causing the subsequent event to occur." [Ref. 4: pp. 43-44] Through the use of a likelihood ratio, Enzer was able to overcome the problems of assymetry and the diminishing effect in the earliest form of cross impact analysis.

2. **Turoff**

Murray Turoff approached the same set of discrepancies in much the same way as Enzer, except that Turoff defined an occurrence ratio, Φ_i, as the natural logarithm of the odds. [Ref. 9] That is,

$$\Phi_i = \Phi(P_i) = \ln O_i = \ln[P_i (1 - P_i)] \qquad \text{(eqn 2.2)}$$

Using this definition, Turoff set about establishing a relationship between the likelihood of occurrence of an event and the effort put forth to enhance or inhibit that occurrence. He wanted this relationship to be "such that if an equal amount of effort is devoted to both enhancing and preventing the occurrence of an event then the likelihood corresponds to a probability of one-half (i.e., random or neutral)." [Ref. 9: p. 319] Furthermore, Turoff assumed that the estimater would provide consistent probability estimates; this was a crucial difference between his approach and those techniques using Monte Carlo simulations. With these assumptions, he computed an expression for P_i, the probability of occurrence of event i:

$$P_i = \frac{1}{1 + \exp(-\gamma_i - \sum_{k \neq i} C_{ik} P_k)} \qquad \text{(eqn 2.3)}$$

where γ_i was a function of unknown variables and C_{ik} was the cross impact term. Turoff then showed that the occurrence ratio could be related to equation 2.3 by using equation 2.2 to get

$$\Phi(P_i) = \gamma_i + \sum C_{ik} P_k$$

He noted that while the impact of the k^{th} event on the i^{th} event was additive with respect to the occurrence ratio, it was multiplicative with respect to odds:

$$O_{ij}(P_j + \Delta P_j) = O_{ij}(P_j) O_{ij}(\Delta P_j)$$

In other words, any change in the probability of occurrence of event j which had an effect on event i would change the odds multiplicatively. He concluded his development with the notion that his series of equations satisfy a likelihood ratio viewpoint of statistical inference since the final odds of an event occuring may be written as the product of an initial odds times a likelihood ratio.

3. **Helmer**

Enzer and Turoff utilize transformations into "odds space" before collectively applying the cross impacts. The inverse transformations always map onto the closed interval from 0 to 1, thus making it impossible to create an illegal probability. Olaf Helmer proposed transforming probabilities, P, into "R-space" to accomplish the same function of mapping (0,1) onto itself in his modification to the cross impact method. [Ref. 5]

Helmer noted that trends have a natural lower bound at 0 and that the maximum natural upper bound of all the trends could be used as the upper bound, U, for all trends. He also needed a "central value", C, which he let equal the median of the trend's estimated value. With these parameters, Helmer defined his transformation as

$$R(P) = K \frac{P - C}{P(U - P)} , \qquad 0 < P < U$$

where K was defined as

$$K = \frac{(C + S)(U - C - S)}{S}$$

and S was a "surprise threshold". The surprise threshold was set such that "we will not be surprised depending on whether the true value turns out to be outside or inside the inter-quartile interval." [Ref. 5: p. 23] In this model, cross impacts are additive in R-space:

$$R(P(i|j)) = R(P) + X(j \text{ on } i)$$

where X(j on i) is the actual cross impact coefficient from the cross impact matrix. Helmer recommended a scale of -3 to +3 for the values of X according to:

$\pm 1/2$ = small	± 2 = large
± 1 = medium	± 3 = overwhelming

Positive and negative values corresponded to enhancing or inhibiting cross impacts as before. Once all impacts had been aggregated, one need only apply an inverse transformation

$$P = \frac{(RU - K) + \sqrt{(RU - K)^2 + 4RC}}{2R}$$

to obtain the final probabilities. A significant difference in Helmer's model and those presented earlier is that impacts last for only one time interval, but this was corrected by the use of a carryover parameter which Helmer described in [Ref. 5].

An excellent comparison between Enzer's likelihood ratio approach and Helmer's R-space approach appears in [Ref. 10]. In this work, Alter does a superb job of summarizing the approaches and conducting a detailed analytic evaluation of the two methods. He evaluates the models based on their internal consistency, robustness, resiliency, fragility, generality, and clarity. Alter's paper should definitely be consulted if there is further interest in either the likelihood ratio model or the R-space model.

4. Bloom

A more straightforward but somewhat restrictive cross impact model was presented by Mitchel Bloom in 1975. [Ref. 11] In this model, the objective was to use the recent past to forecast the near-term future of the system under consideration. Obviously, a prerequisite for the use of Bloom's model is the availability of quantifyable data and a system for study which lends itself to data collection. There are two basic assumptions upon which the model is built:

- The system possesses considerable inertia. The most likely future of the system will be a simple extrapolation of past variable trends. While the probability of this occurring may be small, it is assumed to be larger than any other alternative future.
- The deviation of any trend from its undisturbed future path will have a cross impact on the other trend variables in the system.

There are several reasons why Bloom felt that a deterministic trend cross impact model with a graphical display of system variables was more appropriate than the models presented earlier. First, cross impact models produce a final set of event probabilities; they do not yield a likely scenario that resulted in those probabilities, nor do they provide the user with the dynamic change in the trend variables over time. Second, Bloom noted that, "the cross impact method has not dealt with system trends, although it has been asserted by students of social change that persistence within societal systems, as indicated by trends, is equally important in understanding history as sudden change caused by singular events." [Ref. 11: p. 38] Finally, a trend cross impact model is a way to combine past data with the intuitive perceptions of a group of experts since a graphical picture of the future is often easier for a group to evaluate in terms of its holistic view of the future. With this rationale, Bloom developed an

equation that is easily used by a panel of experts with limited background in the mathematics of forecasting and who have access to relevant data.

Bloom's basic cross impact equation is

$$X_j(t + \Delta t) = X_j(t) + \Delta X_j(1 + \sum_i g_{ij}) \qquad \text{(eqn 2.4)}$$

where $X_j(t)$ is the level of variable X_j at time t, ΔX_j is the extrapolated change in X_j over $(t, t + \Delta t)$ and g_{ij} is the cross impact of trend j on trend i. Equation 2.4 is more easily understood if written in the form:

Cross impacted value of trend j at $(t + \Delta t)$	=	value of trend j at time t	+	a priori increment to trend j	×	change due to the cross impacts of all other trends

There are several points to be made about Bloom's approach. Equation 2.4 expresses the future of a system in terms of persistence, not in terms of change. "It explicitly excludes crises and other system shocks which result in sudden, large, discontinuous changes in the levels of the key variables in the system." [Ref. 11: p. 52] Variables must be viewed as those which change monotonicly. The model cannot deal with trends that are constant over time since this will make ΔX_j equal to zero and implies that cross impacts have on effect. Finally, [Ref. 11] is restricted to only trend-trend impacts. This last deficiency was corrected when Bloom developed an extension to his original model to include events [Ref. 12]. The inclusion of events is accomplished by using the cumulative distribution function (CDF) for the logistics distribution to describe the probability trend of an event over time. Consequently, events could be viewed as trends that increased monotonicly from 0 to some maximum value over a range of time determined by the parameters of the equation. While it is recognized that a true CDF must approach a value of one as t approaches infinity, the equation needed is obtained by multiplying the logistics CDF by a maximum value M.

5. **SMIC-74**

A cross impact model that is substantially different in approach and results is SMIC-74 which was conceived by Duperrin and Godet in 1975. [Ref. 13] They observed that most of the cross impact methods developed up to this point failed to

provide consistent final probabilities[1] and yielded results such as

$$P(i) < P(i|j)P(j)$$

which is incompatible with

$$P(i) = P(i|j)P(j) + P(i|\text{not } j)P(\text{not } j).$$

Duperrin and Godet noted that for a system of n events, there were 2^n combinations of events possible. This could be viewed as 2^n scenarios which corresponded to event i occurring or not occurring for all events from one to n. The most probable scenario usually had a probability of occurrence of only about 0.1, depending on the number of events and the initial probabilities assigned to the events [Ref. 13: p. 304]. However, there were usually quite different scenarios with only slightly lower probabilities than the most probable scenario. Duperrin and Godet felt that for cross impact to overcome this drawback, the method employed would have to produce a rank order of all possible scenarios. The requirement to generate a consistent set of final probabilities and to produce a scenario ranking led to the development of SMIC-74.

The method assumes that a panel of experts will be able to render opinions about:

- The list of n events which are considered to be the key ones for the system under consideration.
- The probability P(i) of each event e_i defined as the probability of the occurrence of e_i within the time period considered.
- The conditional probability of the separate event pairs:

  P(i|j) = the probability of i if j occurs

  P(i|not j) = the probability of i if j does not occur

A given scenario denoted E_k is composed of n separate events; e_i or $ebar_i$ for all events, which correspond to event i occuring or not occuring, respectively. This scenario has an unknown probability π_k and the sum of all π_k must equal one. P*(i) is defined as

$$P^*(i) = \sum_k \theta_{ik}\pi_k$$

where

$\theta_{ik} = 1$ if e_i forms part of E_k

$\theta_{ik} = 0$ if $ebar_i$ forms part of E_k

---

[1]For more details on the issue of consistency in cross imact analysis, see [Ref. 14] and [Ref. 15]

Similar definitions are established for P*(i|j) and P*(i|jbar). These theoretical values of P*(i), P*(i|j) and P*(i|jbar) must satisfy the constraints established by Baysean probability theory. Duperrin and Godet wrote an objective function which was the difference between the P(i|j) estimates from the experts and the theoretical P*(i|j)P*(j) factors expressed in terms of π_k. This objective was minimized subject to the constraints that:

$$\sum \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \text{ for all } k$$

The minimization problem described above is of the quadratic form with linear constraints. Solving this system of equations yields both a consistent set of probabilities and a cardinal ranking of all possible scenarios.

SMIC-74 came under quick criticism by Mitchell and Tydeman [Ref. 16]. They were able to show that the cardinal ranking of scenarios was not unique and further, that the system of equations is potentially very large. If the problem was reformulated in terms of P*(i) and P*(j), and appropriate constraints added, then the system would become a linear programming problem which could identify the multiple solutions. Furthermore, Y. Kaya, et al [Ref. 17: p. 245] show that the use of Dupperin and Godet's quadratic objective requires such computational effort that obtaining a solution is prohibitive if n is very large. Finally, the ability of expert panelists to answer questions that lead to estimates of P(i|j) and P(i|jbar) was doubtful. Mitchell and Tydeman pointed out that "results of studies currently in progress indicate that (1) participants are frequently confused and unsure of the interpretation of such questions and (2) respondents often interpret the questions in terms of time-dependant conditional probability statements." [Ref. 18: p. 133] The significant difference between P(i|j) and P(i|j occured first) is clearly demonstrated in [Ref. 18]. Consequently, SMIC-74, which appears to be a fairly popular cross impact approach has some major difficulties which hamper its effective use.

6. **KSIM**

In 1972, Julius Kane advanced another cross impact technique that was similar to several of the techniques already presented, yet was unique in its approach [Ref. 19]. Unlike other futures researchers, with the exception of Bloom, Kane's primary concern was to develop a model that could be used by people with little mathematical background. He believed that most simulation models were excessively numerical and focused attention on those variables which were easily quantifiable and tended to exclude variables that were basicly subjective in nature. As a result, most

policy makers, for whom these models were designed, were reluctant to invest the time to understand how to use the simulation model. Kane's objective was to attempt to rectify this situation. "It was the purpose of our research to try and design a simulation procedure - or better yet, a simulation language in which technically unsophisticated people could quickly become fluent in the logical expressions of cross impact concepts." [Ref. 19: p. 130] Additionally, Kane sought to structure the problem to allow for a realistic, graphic display of the system variables since this was most easily processed by the user. These notions form the basis of KSIM, Kane's SIMulation language. The model is constructed such that it has the following properties:

- System variables are bounded. With an appropriate set of units these bounds can always be set at 0 and 1.0.
- A variable increases or decreases according to whether the net impact by other system variables is positive or negative.
- The response of a variable to a given impact decreases to 0 as that variable approaches its upper or lower bound.
- A variable will produce a greater impact on the system as it grows larger
- Complex interactions can be broken down into a network of discrete binary interactions.

The KSIM algorithm uses state variables, $X_i(t)$, which are bounded by 0 and 1.0 as described in the first model property. The updated value of $X_i(t)$ is calculated by

$$X_i(t + \Delta t) = X_i(t)^{\pi_i(t)} \qquad \text{(eqn 2.5)}$$

where $\pi_i(t)$ is chosen explicitly as

$$\pi_i(t) = \frac{1 + \tfrac{1}{2}\Delta t \sum_j (|\gamma_{ij}(t)| - \gamma_{ij}(t))}{1 + \tfrac{1}{2}\Delta t \sum_j (|\gamma_{ij}(t)| + \gamma_{ij}(t))} \qquad \text{(eqn 2.6)}$$

and $\gamma_{ij}(t)$ are the cross impacts of variable j on variable i and Δt is the time period of one iteration. The cross impacts are functions of the magnitude of the impacting variable and the rate of change of that variable as shown in equation 2.7.

$$\gamma_{ij}(t) = A_{ij}X_j(t) + B_{ij}(dX_j(t)/dt) \qquad \text{(eqn 2.7)}$$

Equation 2.6 implies that $\pi_i(t) > 0$, thus equation 2.5 always maps the open interval (0,1) onto itself. Equation 2.6 is much clearer if written as

$$\pi_i(t) = \frac{1 + \Delta t[\text{SUM OF NEGATIVE IMPACTS ON } X_i]}{1 + \Delta t[\text{SUM OF POSITIVE IMPACTS ON } X_i]}$$

When the positive impacts are greater that the negative impacts, then $\pi_i(t) < 1$ and results in an increase of $X_i(t)$. Similarly, when the negative impacts outweigh the positive ones, $X_i(t)$ will decrease. This is in accordance with the second assumption property of the KSIM model. To investigate how the remaining properties are satisfied, let $\Delta t \rightarrow 0$. Equations 2.5 and 2.6 can be written as a limiting system of differential equations:

$$\frac{dX_i(t)}{dt} = -\sum_j \gamma_{ij}(t)X_i(t)\ln X_i(t)$$

Kane refers to the term $X_i(t)\ln X_i(t)$ as the "modulator" since as $X_i(t) \rightarrow 1$, $dX_i(t)/dt \rightarrow 0$ and likewise, as $X_i(t) \rightarrow 1$, $\ln X_i(t) \rightarrow 0$ and $dX_i(t)/dt \rightarrow 0$. This is the requirement of the third property. Finally, if the effect of $X_j(t)$ in equation 2.7 is considered individually, it can be seen that it will have a greater effect on $\pi_i(t)$, and ultimately on $X_i(t + \Delta t)$, as the magnitude of $X_j(t)$ increases. Thus, the fourth property is satisfied. Finally, since the system is modelled through the cross impact term, γ_{ij}, which describes a binary interaction between state variables i and j, it can be seen that the final KSIM property holds. The output of KSIM is a plot of each of the X_i variables over time and provides the most concise, yet descriptive means of conveying the results of the simulation.

Lipinski and Tydeman proposed an extension to Kane's model which allows for the inclusion of events as state variables [Ref. 20]. Their extension is similar to Bloom's, in that events are described by their CDF. Since trends in KSIM are described by logistics curves, it seems natural that a logistic CDF would fit nicely into the framework of the model. Lipinski and Tydeman show that if X_i is a trend variable which is initially nonconstant, then updated values of this variable are calulated as

$$X_i(t + \Delta t) = X_i(t)^{Q_i(t)\pi_i(t)}$$

where $\pi_i(t)$ is the cross impact term described previously and $Q_i(t)$ is a function that generates the initial nonconstant trend:

$$X'(t + \Delta t) = X'(t)^{Q(t,\Delta t)}$$

Thus, to include an event in the simulation, the function Q(t) which describes that event's CDF is found and included in the calculation at the appropriate time.

E. EVALUATION

This researcher believes that KSIM, in its extended form, is the most complete and intuitively pleasing model available from the cross impact techniques reviewed. It describes the complex interaction of variables in a realistic, nonlinear fashion, and, once programmed, does not require mathematical sophistication to manipulate the various parameters and interpret the results. The growth characteristics assumed can be modified to accommodate event CDF's of any distribution so the model is extremely flexible. The nature of the model encourages investigating the implications of different policy decisions. Additionally the group interaction that results from evaluating inputs and discussing outputs is often just as valuable as the actual results of the simulation. For these reasons, KSIM appears to be a useful tool for trying to find answers to some of the questions raised in chapter one, and investigate issues surrounding arms transfers. The next chapter will develop, in detail, a KSIM model that can be used to simulate arms transfers.

III. DEVELOPMENT OF AN ARMS TRANSFER MODEL

A. INTRODUCTION

The previous chapter introduced the field of futures research and described the forecasting technique of cross impact analysis. Several improvements to the basic model were discussed, among them KSIM. KSIM is unique in that it is a new mathematical language that facilitates a non-technical decision maker's active involvement in modelling system behavior. This is critical for investigating policy implications in the area of arms transfers since there is very little hard data with which one may construct a simulation model. The purpose of this chapter is to develop an interactive policy simulation based on KSIM to describe the arms transfer process.

B. KSIM PHILOSOPHY

Before proceeding, it is necessary to discuss the underlying philosophy of the simulation. The output of a KSIM simulation is a graphical display of system variables showing how they change over time. As such, KSIM emphasizes the geometry of relationships rather than hard numerical predictions. As an example of what is meant by the geometry of a relationship, consider the following statement, "water is pouring out the hole in a bucket faster than water is being put into the bucket." Without quantifying either flow rate, the size of the hole or the capacity of the bucket, we still have a firm grasp on the general behavior of the system. As Kane, et al, observed, "Subjective evaluations generally correlate well with geometric understanding. If not too much (precision) is asked for it is possible to get more (understanding)." [Ref. 21: p. 66] Thus, while a subjective evaluation is relatively ambiguous by nature, it does contain useful information in the form of geometric relationships. Further, it is in the interpretation and evaluation of geometric relationships that mathematically unsophisticated people can use their intuition and reason, rather than rely on obscure statistical measures and overly precise numerical predictions.

"KSIM calculus is designed to impart a feeling for linkages that cross connect policy variables. As a prime hypothesis we assume that in actual policy implementation, more insight is needed in geometric concepts (the connections between variables, the direction of forces, and the threshold and saturation of variables.) Such

considerations have far more importance than arithmetic specification of parameters. Consequently, KSIM is designed to impart an appreciation of the geometry and dynamics of the system rather than an appreciation of numerical bookkeeping." [Ref. 21: p. 67] With these notions of KSIM philosophy in mind, the relative "softness" of model input requirements seems to make more sense. The required inputs are:

- The list of events and trends considered to be the minimum set of variables that can fully describe the system under study.
- The initial values for these events and trends.
- The strength and mode (enhancing or inhibiting) of the cross impacts between system variables.

With these inputs to the model now defined and the output of the model with its geometric concept described, it is appropriate now to develop an algorithm to relate the two.

C. KSIM MODEL OF ARMS TRANSFERS

As a review, the KSIM model has the following five properties:

- All system variables are bounded
- Variables change according to the net impact of all other variables
- Variable response to net impact approaches 0 as that variable approaches its upper or lower bound.
- Variables will exert a greater impact on the system as the magnitude of those variables grows larger.
- Complex interactions can be broken down into binary interactions.

The mathematical calculations are carried out in an iterative fashion. With Δt being the time interval for one step, future values of system variables, $X_i(t + \Delta t)$, are computed from present values, $X_i(t)$, according to equation 3.1

$$X_i(t + \Delta t) = X_i(t)^{\pi_i(t)} \qquad \text{(eqn 3.1)}$$

The term $\pi_i(t)$ is derived from the cross impacts, $\gamma_{ij}(t)$,

$$\pi_i(t) = \frac{1 + \tfrac{1}{2}\Delta t \sum_j (|\gamma_{ij}(t)| - \gamma_{ij}(t))}{1 + \tfrac{1}{2}\Delta t \sum_j (|\gamma_{ij}(t)| + \gamma_{ij}(t))} \qquad \text{(eqn 3.2)}$$

The cross impact term, $\gamma_{ij}(t)$, is a function of both the present value of the impacting variables $X_j(t)$, and the change in those variables $dX_j(t)/dt$.

$$\gamma_{ij}(t) = A_{ij}X_j(t) + B_{ij}(dX_j(t)/dt) \qquad \text{(eqn 3.3)}$$

Kane et al provide an excellent description of the meaning of the A_{ij} and B_{ij} terms in [Ref. 21]. They observe that the A_{ij} term describes the impact variable j will have on variable i simply because of its existence. For example, the amount of sunshine has this type of impact on plant growth. On the other hand, the elements of the B_{ij} matrix describe the impact that a change in the value of variable j has on variable i. The impact that weather changes often has on arthritis pain is an example of this type of impact. A_{ij} and B_{ij} may be functions of time but are almost always constants. (To require the user to estimate a functional cross impact parameter would be contrary to the concept of simplicity of inputs!)

In its most basic form, the KSIM model is described by these three equations. In this basic form, only trend variables that are initially constant over time may be used. The values of the trend variables are modified according to the cross impacts of other trend variables, so they are not constant in the presence of cross impacts. Lipinski and Tydeman sought to find a way to include trend variables in the simulation that were not initially constant. That is, the trend variables of interest are those that take on new values over time, regardless of the presence of cross impacts. These initial nonconstant variables can be included in the model by first finding the function, $Q_i(t)$, such that it describes the original nonconstant trend, $X_i(t)$, in a recursive way:

$$X_i(t + \Delta t) = X_i(t)^{Q_i(t, \Delta t)} \qquad \text{(eqn 3.4)}$$

Now, to include X(t) in the KSIM procedure the impacts are applied according to equation 3.5:

$$X_i(t + \Delta t) = X_i(t)^{Q_i(t, \Delta t)\pi_i(t)} \qquad \text{(eqn 3.5)}$$

We now note that any given cumulative distribution function (CDF) can be written in iterative form as

$$C_i(t + \Delta t) = C_i(t)^{Q_i(t, \Delta t)} \qquad \text{(eqn 3.6)}$$

where $Q_i(t)$ is a function that will describe the CDF of C_i. Thus, $Q_i(t)$ is a function that yields the a priori increment to the CDF at each time step. Our objective is to be able to learn about the behavior of the system of events that comprise the arms transfer process. One of the difficulties of using the KSIM model is to define exactly what is meant by the loosely used term "system variable". Therefore we shall explicitly define our state variable $S_i(t)$, the value of event i's CDF with due consideration of cross impacts, by the following recursive relationship:

$$S_i(t + \Delta t) = S_i(t)^{Q_i(t, \Delta t)\pi_i(t)} \qquad \text{(eqn 3.7)}$$

where $Q_i(t)$ is the a priori increment to event i's CDF at time t and $\pi_i(t)$ is given by equation 3.2. To initialize the state variable $S_i(t)$, we find the value of time, (t_o), for which $C(t_o) = 0.001$, and let $S(t_o) = 0.001$. In the absence of cross impacts $(\pi_i(t) = 1)$, equation 3.7 is precisely the same as equation 3.6 and the system variable for event i is simply the CDF for i. We shall now explore the properties of our new model.

D. MODEL PROPERTIES

The properties of this new model are examined in terms of Kane's original model. Kane's first property states that system variables are bounded by 0 and 1. Since $Q_i(t)$ generates a monotonically increasing function (the CDF), it must be less than 1 over all t. We are given that $\pi_i(t)$ is greater than 0 by its definition. Since $S_i(t_o) < 1$ by its definition, then the value of

$$S_i(t)^{Q_i(t)\pi_i(t)} < 1$$

Similarly, since $S_i(t_o) > 0$,

$$S_i(t)^{Q_i(t)\pi_i(t)} > 0$$

Thus the property of boundedness in the open interval (0,1) is preserved in the new model.

Kane's second property states that a variable increases or decreases according to whether the net impacts are positive or negative. When there is a net positive impact, $\pi_i(t) < 1$ from equation 3.2. Again, we know that $Q_i(t) < 1$, so the product $Q_i(t)\pi_i(t) < 0$. Thus we can positively state that in the presence of a positive impact, $S_i(t)$ will increase. However if the net impact is negative, $\pi_i(t) > 1$, the behavior of $S_i(t)$ seems to depend on the product $Q_i(t)\pi_i(t)$. Note that when

$$Q_i(t)\pi_i(t) < 1 \qquad S_i(t) \text{ increases}$$
$$Q_i(t)\pi_i(t) > 1 \qquad S_i(t) \text{ decreases}$$
$$Q_i(t)\pi_i(t) = 1 \qquad S_i(t) \text{ remains constant}$$

However, what seems more important is what happens to the state variable in the presence of impacts relative to what happens to the state variable without impacts. To explore this, consider the difference in magnitude between $S_i(t)^{Q_i(t)}$ and $S_i(t)^{Q_i(t)\pi_i(t)}$ for various values of $\pi_i(t)$. Observe that when:

$$\pi_i(t) < 1 \qquad S_i(t)^{Q_i(t)} < S_i(t)^{Q_i(t)\pi_i(t)}$$
$$\pi_i(t) > 1 \qquad S_i(t)^{Q_i(t)} > S_i(t)^{Q_i(t)\pi_i(t)}$$

From the above relationships it can be seen that in the presence of net positive cross impacts, the state variable will increase above its a priori level and when there are net negative impacts, the state variable will be forced below its a priori increment. Therefore, we can conclude that, in terms of our model, a property similar to Kane's second property is satisfied.

The third property is that a state variable's response to the cross impacts of other system variables will decrease to 0 as that state variable approaches its upper or lower bound. To investigate this property, we take the derivative of S_i with respect to time:

$$\frac{S_i(t + \Delta t) - S_i(t)}{\Delta t} = \frac{S_i(t)[S_i(t)^{(Q_i(t)\pi_i(t) - 1)} - 1]}{\Delta t}$$

From this expression, it can be seen that as $S_i(t)$ approaches 0, $dS_i(t)/dt$ approaches zero. Also, when $S_i(t)$ approaches 1, $[S_i(t)^{(Q_i(t)\pi_i(t) - 1)} - 1]$ approaches 0, so $dS_i(t)\ dt$ still goes to 0. Clearly, Kane's third property is applicable to the new model.

Kane's fourth property is that, all other things being equal, a variable will exert a greater influence on the system as its magnitude increases. Equation 3.3 states that $\gamma_{ij}(t)$ is a function of both the magnitude of $S_j(t)$, and the rate of change of $S_j(t)$. Thus, as the magnitude of $S_j(t)$ increases it will produce a larger value of $\gamma_{ij}(t)$ and, consequently a larger net impact on state variable i. The fourth KSIM property holds.

Finally, the fifth assumption is that only binary interactions may be considered. This property holds since system behavior is still modelled through the cross impacts given by $\gamma_{ij}(t)$, which are solely a function of event pairs.

Thus, our new model for the value of an event CDF with consideration of cross impacts is truely a KSIM type cross impact procedure. As such, we are not bound by the necessity to find a suitable numerical interpretation for our state variable, $S_i(t)$; our primary concern is in the geometry of the system variables and the effects they have on each other under various circumstances. The model has utility regardless of our ability to make probability statements from the values of its state variables at any particular time, since information conveyed in terms of probabilities is much less appealing than a visual display of the growth rates and magnitudes of the state variable over time. Again we refer to the basic philosophy of the KSIM procedure: "The significant difference in orientation between our procedures and most other methods is that we emphasize the geometry of the system, the structural relationships between the variables while standard procedures tend to emphasize arithmetic details, the precise specification of coefficients and parameters." [Ref. 22: p. 286]

Now we shall take up the task of implementing this model. The first part of this task consists of two elements; the development of an expression for $Q_i(t)$ and providing some method to estimate the parameters needed to specify $Q_i(t)$. The rest of this chapter addresses the details of this first task.

E. DETERMINATION OF Q(T)

The arms transfer process can be broken down into eight discrete events. The question is what type of CDF should we use to model these events? Mitchell Bloom makes a strong case for using the logistics equation to describe the CDF of an event. [Ref. 12: p. 185]

The probability density function (PDF) for the logistics distribution is an inverted U-shape (Figure 3.1). The interpretation of this PDF is that there is a negligible probability of occurrence at early times which rises to a maximum and falls back off to a negligible level. "The concept of representing event probability densities in this

manner is neither strange nor new. It is often said that if an event does not occur before a certain point in time, its probability declines rapidly until it is almost certain not to occur. Examples include: waiting for a friend to arrive at a predetermined place and time or waiting for the party one has called to answer the telephone." [Ref. 12: p. 185] This notion is fully compatible with describing arms transfer events; an order for a weapon system has a maximum probability of occurrence some days after a negotiation and if the order is not placed within a certain period of time, it probably will not occur. For these and for reasons of mathematical tractability in the KSIM procedure, the logistics curve[2] will be used to model arms transfer event CDF's.

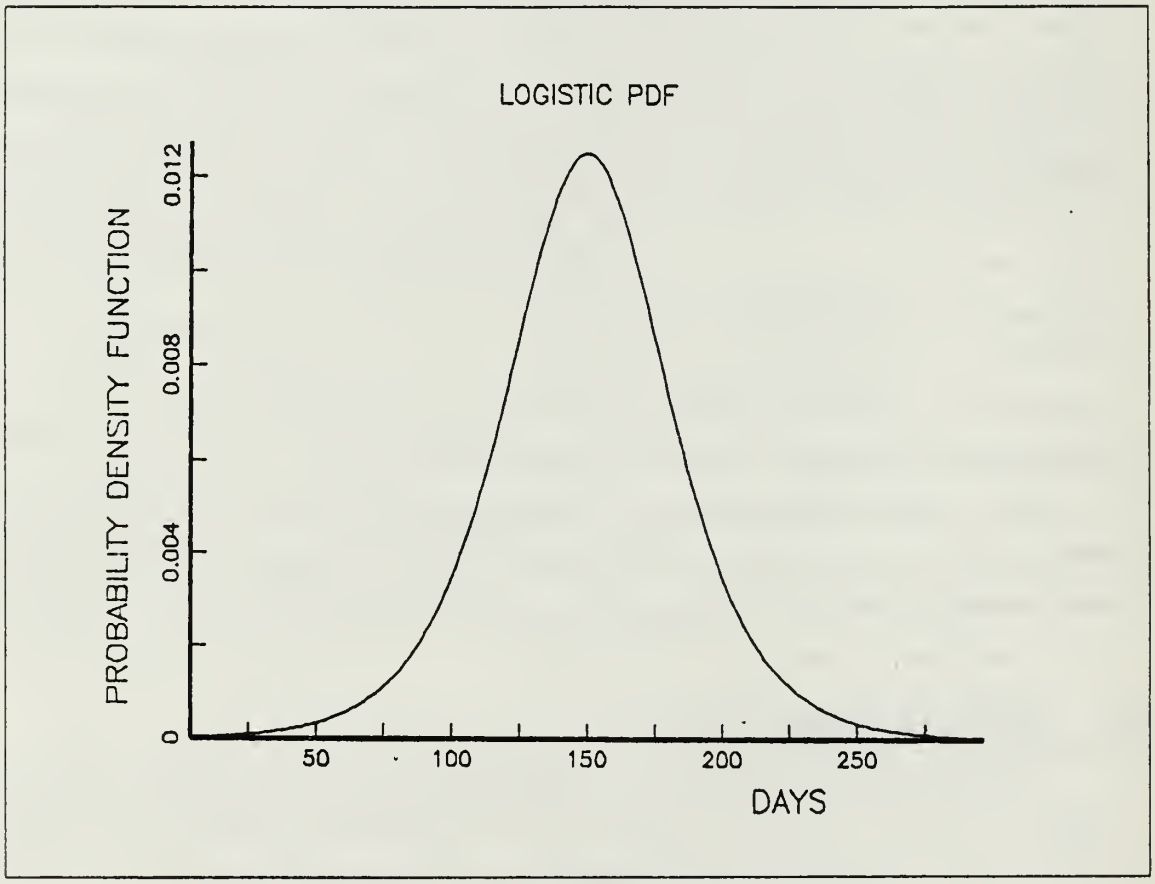

Figure 3.1 Logistics Curve PDF.

[2]If it is later decided that some other distribution is more appropriate, the technique used to find $Q_i(t)$ may be used on any differentable function.

1. **Event Parameter Equations**

The logistics CDF is given by equation &logeq.. (For ease of notation and understanding, the remainder of this development will consider only one event and will omit its subscript, i. When a final computational form is reached, the subscript will be replaced.)

$$C = \frac{1}{1 + \exp(-\alpha t + \beta)} \qquad \text{(eqn 3.9)}$$

where C is the value of the CDF at time t, and α and β are constants to be determined.

To compute α and β, we estimate the time at which the cumulative probability is equal to 0.5 denoted t_h. The time at which the cumulative probability is equal to 0.1 will be denoted t_e. (See Figure 3.2) Thus,

$$\text{when} \quad t = t_h \;,\; C = 0.5$$
$$t = t_e \;,\; C = 0.1$$

Equation 3.9 can be written in logarithmic form:

$$-\alpha t + \beta = \ln[(1/C) - 1] \qquad \text{(eqn 3.10)}$$

Substituting the values of C for t_h and t_e shown above into equation &lnlogcdf and solving for α and β yields:

$$\alpha = [1/(t_h - t_e)]\ln 9 \qquad \text{(eqn 3.11)}$$

$$\beta = [t_h/(t_h - t_e)]\ln 9 \qquad \text{(eqn 3.12)}$$

Now, in order to obtain the function Q(t) from the iterative equation 3.6, we will take the derivative of C with respect to t:

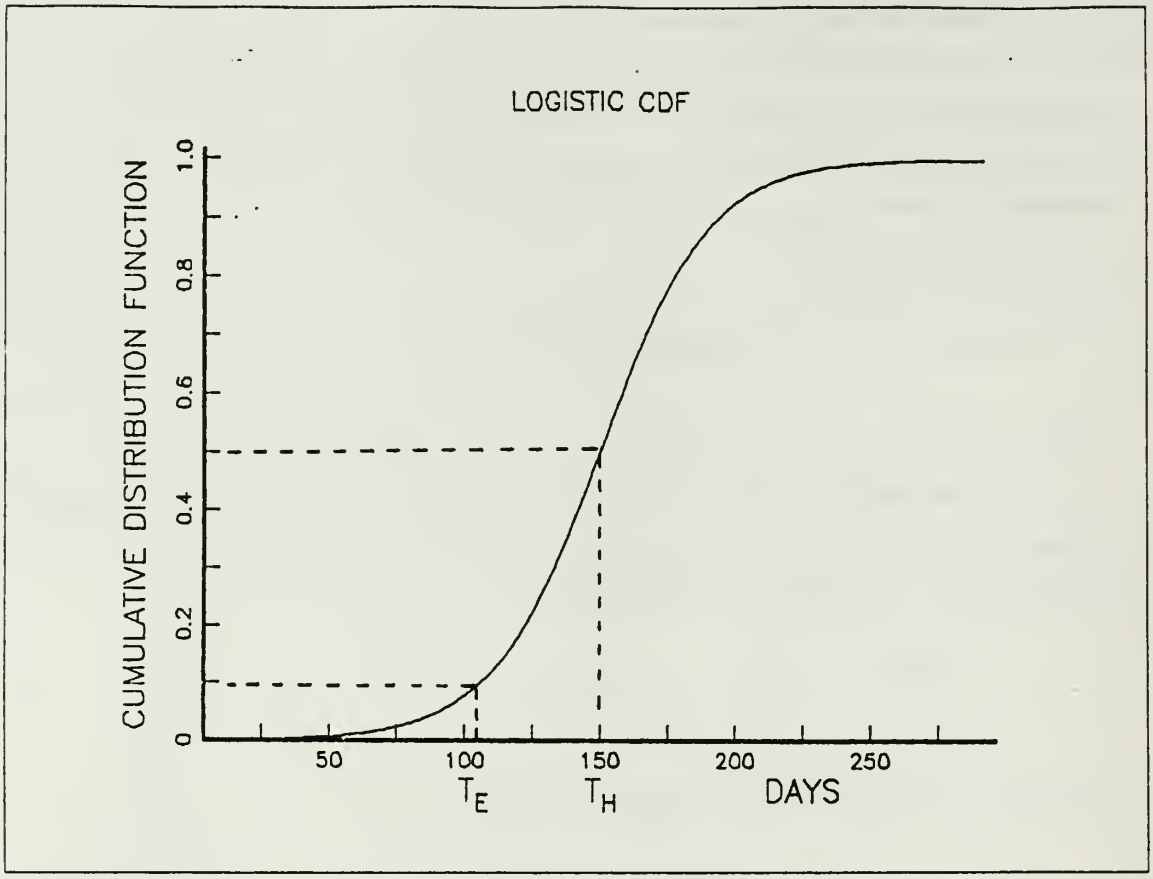

Figure 3.2 Logistics Curve CDF.

$$\frac{dC(t)}{dt} = \frac{\alpha \exp(-\alpha t + \beta)}{(1 + \exp(-\alpha t + \beta))^2} \qquad \text{(eqn 3.13)}$$

As an approximation, set

$$\frac{dC(t)}{dt} = \frac{C(t + \Delta t) - C(t)}{\Delta t} \qquad \text{(eqn 3.14)}$$

Setting equations 3.14 and 3.13 equal to each other and solving for $C(t + \Delta t)$ results in

$$C(t + \Delta t) = C(t)\left[1 + \frac{\alpha(\Delta t)\exp(-\alpha t + \beta)}{1 + \exp(-\alpha t + \beta)}\right] \qquad \text{(eqn 3.15)}$$

Since we want to get out of the recursive equation and find a closed form solution for Q(t), we will set the right hand side of equation 3.15 equal to $C(t)^{Q(t)}$:

$$C(t)^{Q(t)} = C(t)\left[1 + \frac{\alpha(\Delta t)\exp(-\alpha t + \beta)}{1 + \exp(-\alpha t + \beta)}\right] \qquad \text{(eqn 3.16)}$$

Finally, taking the natural logarithms of equation 3.16 yields equation 3.17.

$$Q(t)\ln C(t) = \ln C(t) + \ln\left[1 + \frac{\alpha(\Delta t)\exp(-\alpha t + \beta)}{1 + \exp(-\alpha t + \beta)}\right] \text{(eqn 3.17)}$$

By dividing equation 3.17 by lnC(t), we have an expression for Q(t). The subscripts are now returned and our final closed form equation for $Q_i(t)$ is

$$Q_i(t) = 1 + \frac{\ln\left[1 + \frac{\alpha_i(\Delta t)\exp(-\alpha_i t + \beta_i)}{1 + \exp(-\alpha_i t + \beta_i)}\right]}{\ln\left[\frac{1}{1 + \exp(-\alpha_i t + \beta_i)}\right]} \qquad \text{(eqn 3.18)}$$

F. EVENT PARAMETER ESTIMATION

In order to implement this model the user will be required to supply estimates for the following parameters for each event:

- Time when CDF is equal to 0.5, t_h
- Time when CDF is equal to 0.1, t_e
- Strength and mode of the existance of this variable on every other variable
- Strength and mode of an increase in this variable on every other variable.

If the user is very familiar with the arms transfer process, estimation of these values may not prove to be difficult. As John Mather points out though, "A more useful development would be the provision of some means of estimating the initial values of the (events) from available data." [Ref. 23: p. 21] If a first estimate of event parameters could be provided by an existing data base, then this would greatly reduce the burden on the analyst and provide a solid basis for discussions regarding the acceptance or alteration of the estimated parameters. Third Point Systems, Inc., Monterey,

California has made a substantial effort to record arms transfer data for almost every country in the world. It is possible to extract event parameter estimates (with the exception of the cross impacts) from this data set.

1. **Event Data Set**

The Arms Transfer Data Set was developed by Third Point Systems to record individual arms transfer events which could be aggregated to describe the entire arms transfer process. "The purpose for the creation of the data set was to aid foreign policy decision makers to evaluate the patterns, purposes and effects of international arms transfers." [Ref. 24: p. 5] The other major sources of arms transfer data, such as SIPRI and ACDA, have concentrated only on the value of the weapons transferred, and the date the transfer took place. Adding further to the confusion, these sources use different dates for the actual arms transfer; some use the date the contract is signed, some use the date the order is placed, and others use the actual date of delivery. However, an arms transfer consists of many discrete events and focusing only on the cash value of a weapon and its date of transfer severly limits a thorough analysis of the arms transfer process. Recognizing this unneccesary limitation, Third Point Systems began to build an extensive data set that recorded each arms transfer event. With this data, the analyst could interpret the subtle changes in the attitudes of various nations towards weapons sales and procurement. Having done this, the foreign policy analyst might be better prepared to attempt to influence other country's actions and provide a small measure of control over the arms transfer process.

Third Point Systems identified fifteen distinct events that comprised the various stages of an arms transfer. For the purposes of this study, some of these events, such as capture, were eliminated as not being particularly influential on the arms transfer process. Other events, such as reject/refuse (recipient) and reject/refuse (supplier), were combined into one event since they were not different enough to be considered separately. Thus, the event set used in this analysis consists of the following eight events (the descriptions are based on the Arms Transfer Handbook [Ref. 24):]

- **Meet/Visit.** This event occurs when two countries meet to consider an arms transfer issue. It includes the exchange of notes, messages, and information as well as the actual meeting between members of the two countries.
- **Propose/Request.** When a supplier country offers a weapons system or support package, or when the recipient directs a request to the supplier, then the event is coded as a Propose/Request.

- **Evaluate/Negotiate.** Events in this category indicate that a country is in the process of considering the purchase of some system. Evaluations can be both a supplier country considering a sale or a recipient country considering a purchase. Also, agreements between industries or governments to produce a specific piece of equipment for development is considered an Evaluation.
- **Reject/Refuse.** This event occurs when either the recipient or supplier reacts negatively and rejects its counterpart's offer or proposal.
- **Order.** If a contract is signed or awarded, or if an agreement is made to purchase or coproduce, then the event is coded as an Order.
- **Delivery.** This type of event includes the actual delivery of a weapons system, the return to an operational state after overhaul by an external supplier, and the licensed production or coproduction of weapons systems.
- **Increase.** Decisions to resume a previously halted delivery, reduce sanctions or supplement a previous order are all considered Increases.
- **Withdraw/Cancel.** Events in this category are those where the supplier or recipient slows production, reduces the quality of weapons, or cancels an agreement.

Figure 3.3 shows the relationship between these events and possible linkages that exist between them.

2. **Data Set Coding**

The event data record is made up of two parts; the first is an analytical section with a strict coding scheme, and the second is a narrative summary of the event. The first section contains the data which can be used to provide estimates of event parameters. This section will be discussed in detail. A sample coding of an arms transfer event appears in Table 1.

The analytical section of the data will now be broken down line by line.

Line 1 Field 1: Month (1)
Field 2: Day (31)
Field 3: Year (86)

The information contained on line 1 is self explanatory. Event dates are only coded when the event occurs; if an event is predicted, it is coded as a Comment. Similarly, if an event is alluded to which occurred more than six months prior to the source date, it is not coded; it is assumed that these events were coded earlier.

Line 2 Field 1: Actor country (ALG = Algeria)
Field 2: Domestic actor (GVT = government)
Field 3: Event code (53 = Request/Inquire)
Field 4: Domestic target (GVT = government)

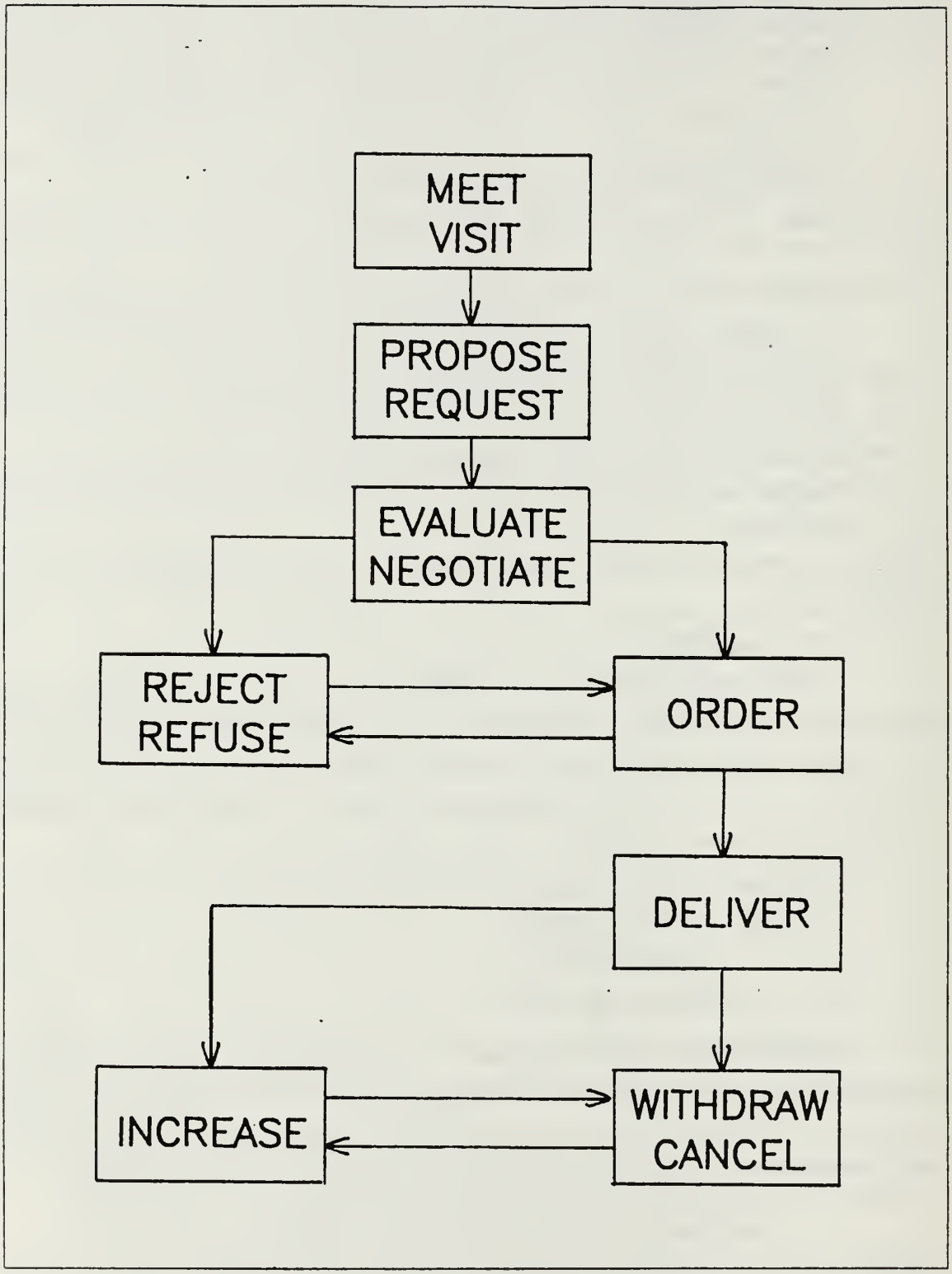

Figure 3.3 Arms Transfer Process.

TABLE 1
EVENT DATA RECORD

```
  1          31          86

ALG        GVT     53       GVT  BRA

  35

EE-9                          AVREWH EI


Algeria shows interest in the Engesa Cascavel
(Brazil) EE-9 armored combat vehicle with a
90mm gun.
```

Field 5: Target country (BRA = Brazil)

An actor country is the country who initiates an arms transfer event. The actor country can be either the supplier or the recipient; the key being which country initiated the action. For example, in a Request, the recipient is the actor, whereas in an Offer, the supplier is the actor. The target country is the object of the action initiated by the actor country. The domestic actor and the domestic target refer to the person, organization or group within the respective country that is responsible for the event. If no domestic actor/target is specified, the government is assumed. The event code refers to the fifteen specific events that comprise the arms transfer process.

Line 3 Field 1: Terms of sale (35 = other)

The terms of sale of a weapons system is indicated if the information is available. Examples of these terms include coproduction, offsets, gifts and credit.

Line 4 Field 1: Weapon system (EE-9)
Field 2: Equipment type (AVREWH EI)

The weapon system is the broad category of hardware/support. The equipment type uses a code to specifically break down the system to the component being transfered.

After the data is collected, it is assembled in the form of "storylines" for review. The storyline simply collects all events for a given country both when it is an actor and when it is a target. In this way, the reviewer can see if there exists a coherent, logical progression of events. Once the reviewer is satisfied with the recorded data, it is loaded into a mass storage device.

3. **Parameter Estimation Technique**

To obtain an estimate for t_h (the time when the CDF is equal to 0.5) and t_e (the time when the CDF is equal to 0.1), one must find a series of events between two countries that deals with the same weapon system. For example, the offer of the United States to sell F/A-18's to Israel, the evaluation by Israel and the subsequent order, etc. might be used as a source of one piece of data. The number of days between event pairs is recorded for all storylines that exist in the data set for similar weapons. (Presumably it takes longer to evaluate the purchase of a squadron of F/A-18's than a load of M-16 rifles.) A CDF would then be constructed from this data. The day when the CDF was equal to 0.5 would be t_h and the day when the CDF was equal to 0.1 would be t_e. This approach requires a great deal of data since the number of storylines in the data set is relatively small, however, this analyst only had access to a small portion of the data set, so an extensive study in this field was not possible. (Indeed, this type of study would be expansive enough to support an entirely separate research report!) In the data provided, seven samples were found that contained the linkage between Evaluation/Negotiation and Order on similar pieces of equipment. The parameter estimation technique will be demonstrated with these seven data points. The raw data are shown in Table 2. These seven data elements were found in storylines from various countries. Obviously, to implement this technique, one would restrict themselves to the arms transfer data from the country of interest.

To construct the CDF, we compute the probability that the number of days between an evaluation and order, a random variable T, will be less than or equal to the various elapsed times, t, from Table 1 That is,

$$\Pr(T \leq t)$$

The results of this calculation are shown below.

t	27	60	64	66	75	89	231
$\Pr(T \leq t)$	.143	.289	.429	.571	.714	.857	1.000

TABLE 2
DATA FOR ESTIMATION OF EVENT PARAMETERS

Date of Evaluation/Negotiation	Date of Order	Elapsed Time (T)
09/02/85	11/16/85	75
08/01/85	09/30/85	60
10/28/85	12/31/85	65
02/08/86	03/06/86	27
10/05/85	12/10/85	66
01/20/86	04/19/86	89
07/01/85	02/17/86	231

Therefore, we would use 66 as our estimate of t_h and 27 as our estimate of t_e. At this point, we recognize that with all eight event CDF's going from 0 to 1, the plot will get cluttered very quickly. Further, we know that by multiplying the CDF by a constant does not ruin any of the five properties of the model previously discussed. We could arbitrarily select values for these constants since they have no numerical meaning in our model, but to give one more piece of visual data, we will use the relative frequency of each event in the Third Point data system and denote this constant M.

$$C = \frac{M}{1 + \exp(-\alpha t + \beta)} \qquad \text{(eqn 3.19)}$$

As an example, suppose Saudi Arabia was the country in question. Table 3 summarizes Saudi Arabia's arms transfer activity for the year 1985 -1986. From this summary it can be seen that the relative frequency of Order is 0.474, so this would be the value of M.

TABLE 3

SAUDI ARABIA ARMS TRANSFER ACTIVITY (1985 - 1986)

Event	Nr of Occurances	Percentage
Propose/Offer	16	5.8
Request	17	6.2
Evaluation/Negotiation	70	25.5
Reject/Refuse (supplier)	3	1.1
Reject/Refuse (recipient)	3	1.1
Order	130	47.4
Deliver	13	4.7
Increase	17	6.2
Withdraw/Cancel	2	0.7

As stated earlier, the data necessary to make reasonable estimates for all event parameters was unavailable. The values of the parameters used in further analysis are based on the author's estimates and in no way are intended to represent any one country or arms transfer.

G. KSIM ALGORITHM

To be effective as a policy analysis tool, the KSIM model for arms transfers must be programmed; the calculations, while not difficult, are too numerous to be handled manually. Additionally, the KSIM program should be portable so that it is not bound to a main frame facility and become relatively inaccessible to the intended user. For these reasons, it was decided to program the KSIM model on an IBM XT personal computer. The program follows the algorithm outlined below:

Step 1. Input the event parameter estimates t_h, t_e and M for all events.

Step 2. Input the cross impact matrices.

Step 3. Compute the values of α and β according to equations 3.11 and 3.12.

Step 4. Compute the time when the event cumulative probability is equal to 0.001. If this time is less than 0, then compute the value of the cumulative probability at time 0. (Use equation 3.19).

Step 5. Determine the start time of the simulation. The start time will be the minimum of each event's t_o or, if this is less than 0, the start time will be 0. Set the simulation time equal to the start time.

Step 6. Check all events to see if their t_o is less than or equal to the current simulation time. Any such events become part of the simulation with an initial value of 0.001.

Step 7. Compute γ_{ij} according to equation 3.3.

Step 8. Compute π_i according to equation 3.2.

Step 9. Compute $S_i(t + \Delta t)$ according to equation 3.7.

Step 10. Repeat **Step 6** until the stop time (selected by the user) is reached.

This algorithm was programmed using Turbo Pascal (version 3.01a). A listing of the source code appears in the Appendix. To make the program functional as an analysis tool, it had to be user friendly, interactive and allow various parameters to be changed. As such, the program is menu driven, and allows data editing of event parameters and cross impact values. Upon termination of the session, the current values of all event parameters and the cross impact matrix are stored on the disk. When the program is loaded, it recalls these parameters from the disk and returns the system to its state at the end of the previous session. The next chapter will show some results from this program and discuss how the model might be used to investigate various arms transfer policy decisions.

IV. RESULTS AND CONCLUSIONS

A. MODEL RESULTS

1. Input Parameters

As stated in the previous section, the data used in the testing of the KSIM arms transfer model were estimated by the author. Table 4 shows the initial event parameters used in the simulation. The values for the relative frequency of occurrence, M, came from Table 3. Since there was no value for the event Meet/Visit, 0.01 was used. The two events, Propose/Offer and Request, were lumped together to form one event, Propose/Request (see the subsection on Event Data Set). The value of M for Propose/Request, then, is the sum of the values for Propose/Offer (0.058) and Request (0.062).

TABLE 4
INPUT PARAMETERS FOR KSIM SIMULATION

Event	t_h	t_e	Relative Frequency
Meet/Visit	20	10	.0100
Propose/Request	35	15	.1201
Evaluate/Negotiate	55	40	.2554
Reject/Refuse	60	50	.0220
Order	80	60	.4744
Deliver	100	85	.0474
Increase	130	100	.0602
Withdraw/Cancel	140	120	.0070

The first run of the model used a value of 0 for all A_{ij} and B_{ij}. This implies that there are no cross impacts and the results should yield a "truncated" CDF; the value of the CDF multiplied by its relative frequency. As can be seen from Figures 4.1 and 4.2, this is indeed the case.

The cross impact matrices were then estimated using Figure 3.3. If there was a line connecting two events, it was assumed that an increase in the probability of occurrence of the first will have some effect on the probability of the second. Recall that positive values are enhancing impacts, while negative values are inhibiting. The scale used for the derivative (B_{ij}) impacts was:

0 = None
1 = Mild
2 = Strong
3 = Overwhelming

Table 5 shows the values for the derivative cross impact matrix that were used in the simulation.

The data for the constant cross impact matrix (A_{ij}) was estimated in a similar manner. Recall that this matrix is the impact that the first variable has on the second simply because of its existence. For this matrix, however, the following scale was used:

0.0 = None
0.1 = Mild
0.2 = Strong
0.3 = Overwhelming

Table 6 displays the values used for the constant cross impact matrix:

The data were entered into the model and a period of 200 days was simulated. The results of this first run appear in Figure 4.3. The output shows that the cross impacts have a great influence on the events Evaluate/Negotiate , Reject/Refuse and Order. These three event state variables reached their upper bounds even though their final values without cross impacts was very low. To understand why, we must closely examine the cross impact matrices and couple the values we see in the tables with the behavior we observe on the plot. As an example, let us try to determine why the state variable for Reject/Refuse grew so rapidly and to such a large magnitude.

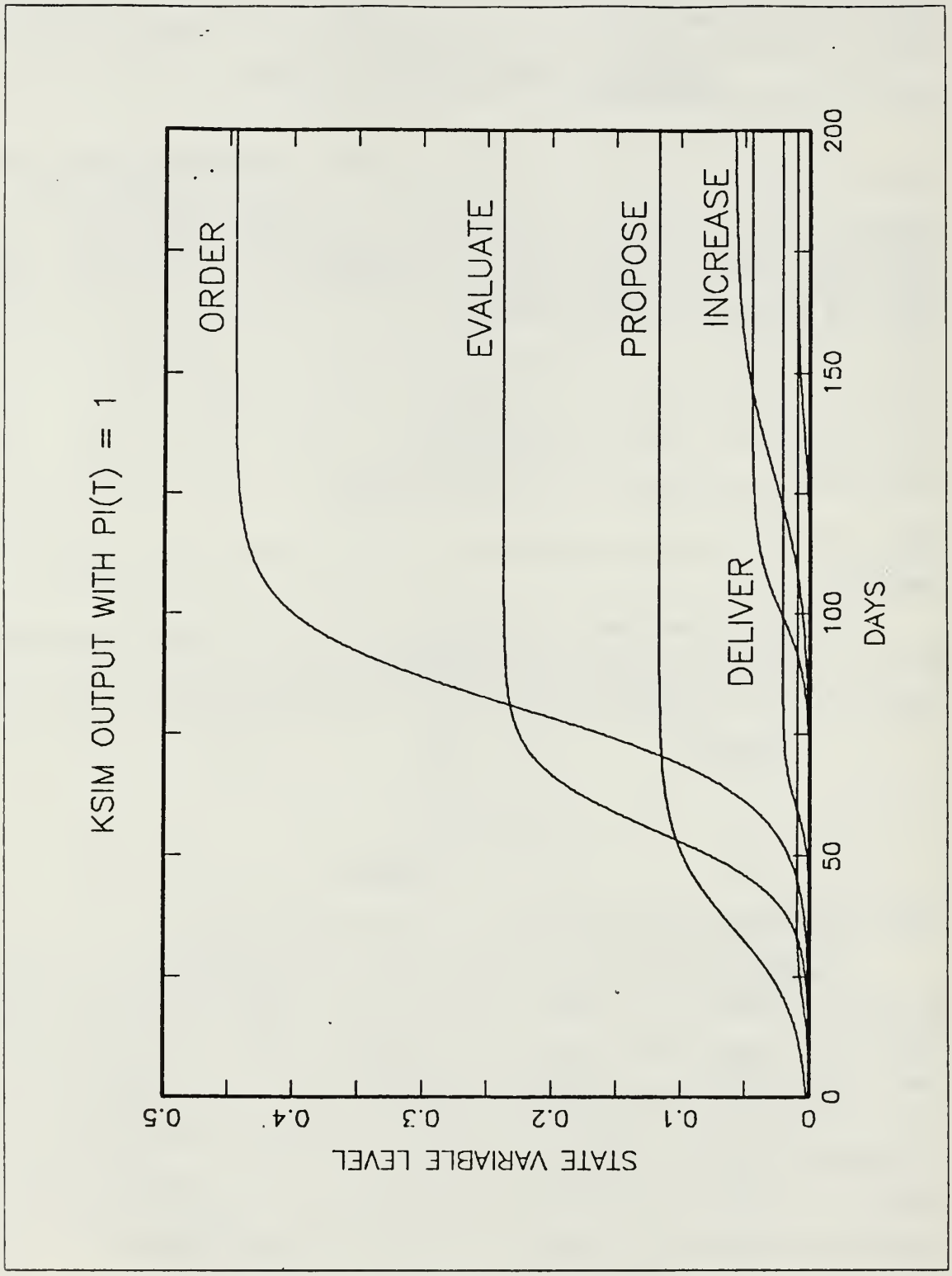

Figure 4.1 KSIM Simulation With No Cross Impacts (0 - 0.5).

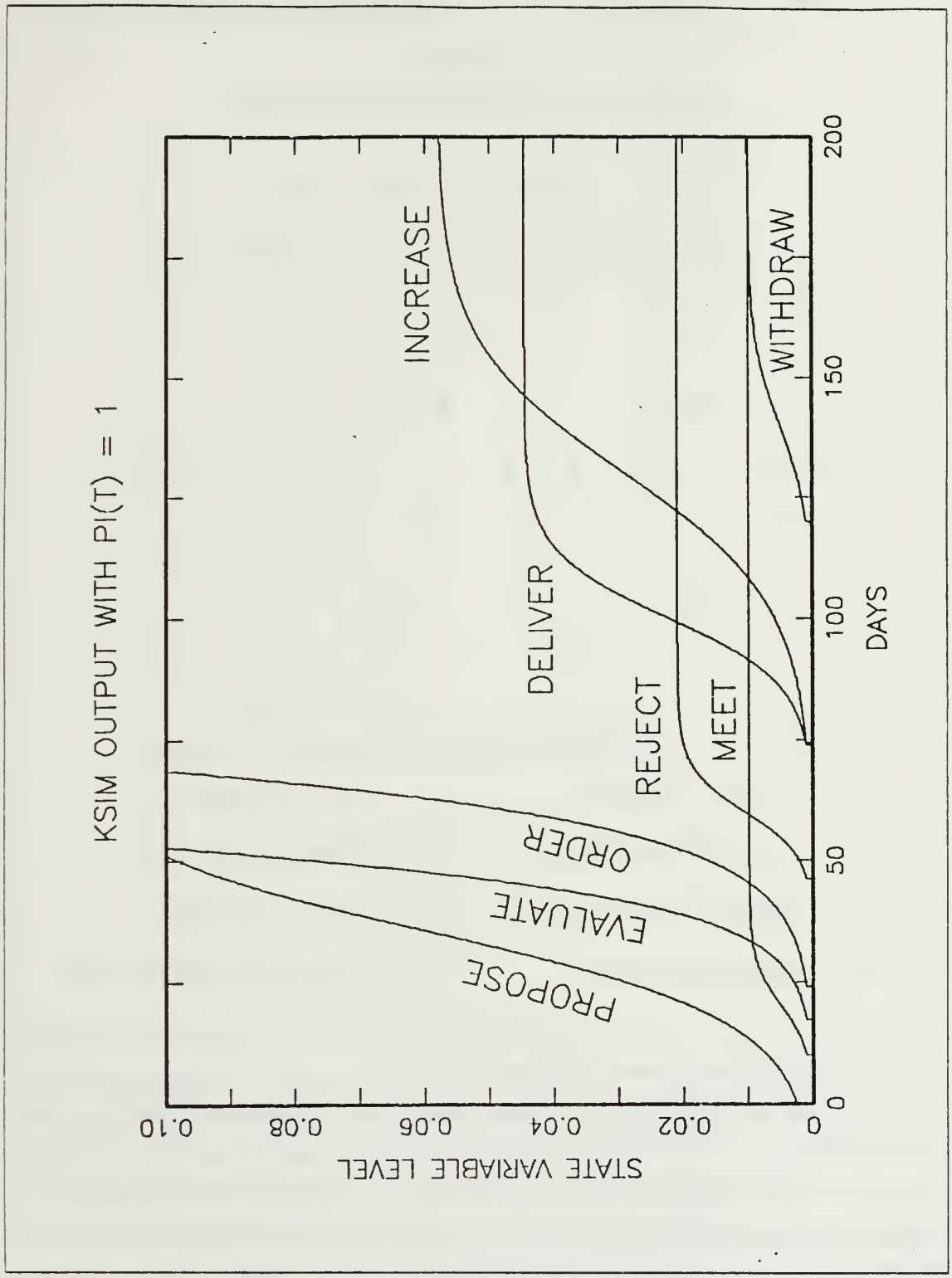

Figure 4.2 KSIM Simulation With No Cross Impacts (0 - 0.1).

TABLE 5

DERIVATIVE CROSS IMPACT MATRIX

		IMPACTING EVENT							
	B_{ij}	1	2	3	4	5	6	7	8
	1	0	0	0	0	0	0	0	0
	2	1	0	0	0	0	0	0	0
	3	0	3	0	0	0	0	0	0
IMPACTED	4	0	0	2	0	-2	0	0	0
EVENTS	5	0	0	1	-3	0	0	0	0
	6	0	0	0	0	2	0	0	0
	7	0	0	0	0	1	1	0	-2
	8	0	0	0	0	1	1	-2	0

Event 1: Meet/Visit

Event 2: Propose/Request

Event 3: Evaluate/Negotiate

Event 4: Reject/Refuse

Event 5: Order

Event 6: Deliver

Event 7: Increase

Event 8: Withdraw/Cancel

Refer to Tables 5 and 6. Reject/Refuse has a positive derivative impact from Evaluate/Negotiate and a negative impact from Order. Thus, when there is a positive slope in Evaluate/Negotiate, the incremental increase in the state variable will enhance the growth of Reject/Refuse. Similarly, as the state variable for Order increases, it will inhibit the growth of Reject/Refuse. Now since the B_{ij} values are the same (2) the net derivative impact will depend on the difference in slope between the two impacting variables. It can also be seen that the state variables for Propose/Request and

TABLE 6

CONSTANT CROSS IMPACT MATRIX

		IMPACTING EVENT							
	A_{ij}	1	2	3	4	5	6	7	8
	1	0	0	0	0	0	0	0	0
	2	.1	0	0	0	0	0	0	0
	3	0	.2	0	0	0	0	0	0
IMPACTED	4	0	.1	.3	0	0	0	0	0
EVENTS	5	0	.1	.3	-.2	0	0	0	0
	6	0	0	0	-.2	.3	0	0	0
	7	0	0	0	-.2	0	.1	0	-.1
	8	0	0	0	.1	.1	.1	-.1	0

Event 1: Meet/Visit

Event 2: Propose Request

Event 3: Evaluate/Negotiate

Event 4: Reject/Refuse

Event 5: Order

Event 6: Deliver

Event 7: Increase

Event 8: Withdraw/Cancel

Evaluate/Negotiate have positive constant impacts on Reject/Refuse. Whenever Evaluate/Negotiate and Propose/Request have any magnitude, they will enhance the growth of Reject/Refuse. Now look at Figure 4.1. We can see that both Propose/Request and Evaluate/Negotiate have rapidly increasing state variables (large slope)and that their growth is both faster (higher slope) and of greater magnitude than Order up to day number 75. During the period from day 0 to 75 then, Reject/Refuse has a net positive impact that accelerates its growth. In fact, the growth rate is so high

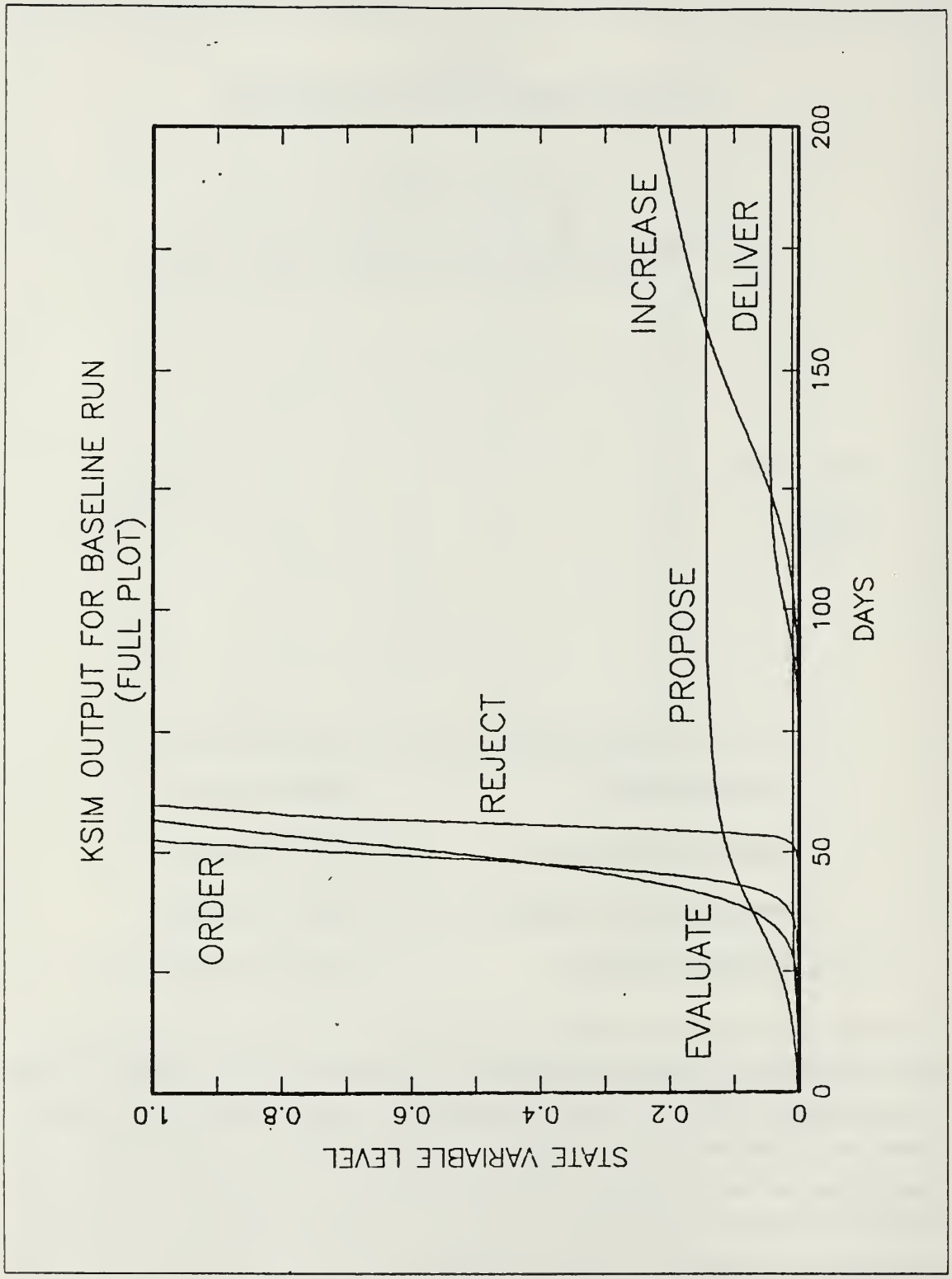

Figure 4.3 Baseline Run of KSIM Model (Full Plot).

that Reject/Refuse has reached its upper bound before the negative impact from Order can have any effect. One can explain the growth or decay of any variable through a similar line of reasoning.

In order to demonstrate the effect that varying the values of the cross impact parameters will have on the system, let us suppose that the A_{ij} matrix did not correctly reflect our view of the way that system variables interacted. We believe that Order would have an overwhelming negative constant impact on Reject/Refuse. To see the effect, we simply set A_{54} equal to -0.3 and run the model. The results are shown in Figure 4.4. The plot shows that the impact of Order on Reject/Refuse has a great influence on the system, since it not only suppresses the growth of Reject/Refuse but also has a secondary effect of enhancing the state variable Withdraw/Cancel. The ability to quickly spot such unexpected secondary effects is one of the excellent qualities of the KSIM procedure.

Let us now consider the effect of a large positive impact on a state variable. It may be our belief that several events have a positive constant cross impact on Deliver, so we shall set A_{16} (Meet/Visit on Deliver) equal to 0.1, A_{36} (Evaluate Negotiate) equal to 0.2, and change A_{46} (Reject on Deliver) from -0.2 to 0. The model is run for a 200 day simulation and the results appear in Figure 4.5. Again we see that the net positive impacts did have a marked effect on the state variable Deliver, but the secondary effect of greatly enhanced growth of Increase is even more pronounced. The caused of this secondary effect can be traced to the entry B_{67}: the increasing value of Deliver has a positive derivative impact on the state variable Increase. We continue to modify the cross impact parameters, observing the effect of these changes until we obtain a set of entries in both of the cross impact matrices that result in system behavior that appears correct. It is during this process of refining the model that the analyst will learn a great deal about the interplay of state variables within the system. Many users of the KSIM procedure state that this acquired knowledge is often of more value than the results of various simulation runs. It will be assumed that the state of the model as specified by the cross impact matrices in Tables 5 and 6 is satisfactory at this point, and proceed with investigating some of the arms transfer policies suggested in chapter one.

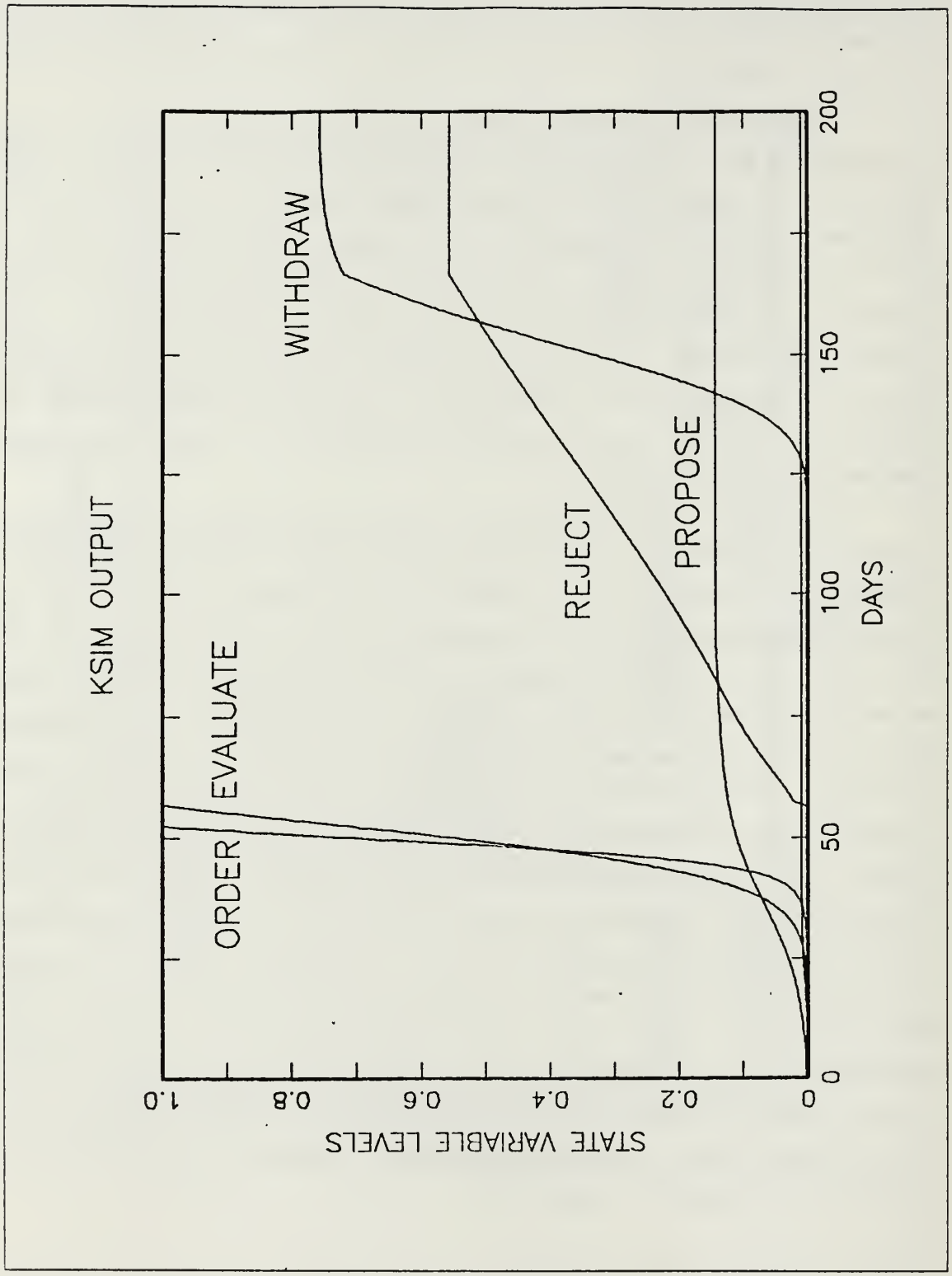

Figure 4.4 Effect of a Large Negative Impact of Order on Reject/Refuse.

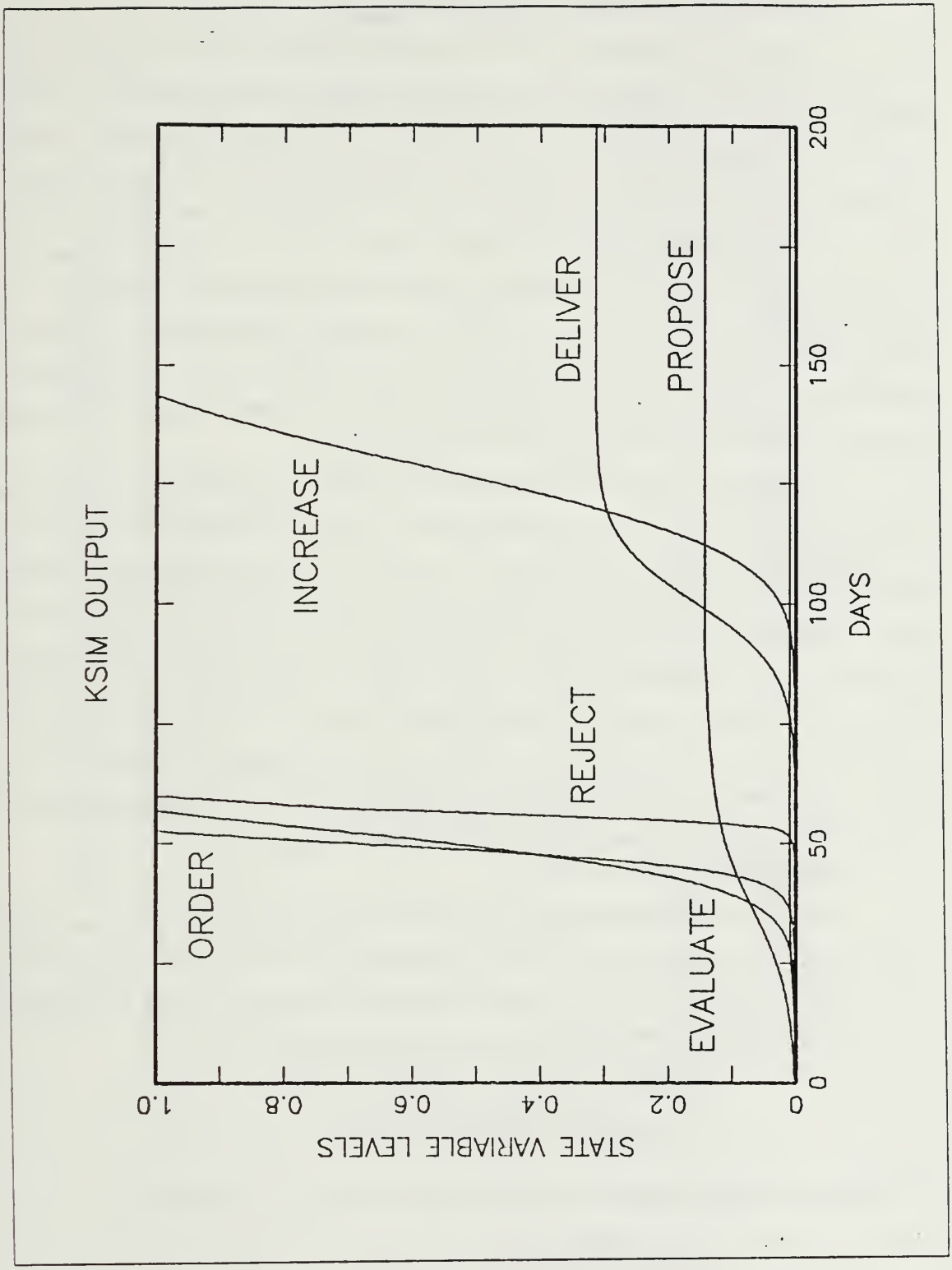

Figure 4.5 Effect of a Large Positive Impact on Deliver.

2. **Promoting the Transfer of Defensive Weapons**

Chapter one alluded to the fact that some weapons lead to greater political stability than others by their technical nature and defensive mission, and that it may be desirable to promote the transfer of these weapons. One policy to achieve this goal might be to actively "sell" the weapon to target countries by demonstrating its capabilities and proposing its purchase. We can now check to see if this policy will have the desired effect. To do this, the value of M, the relative frequency of occurrence, will be changed from 0.1201 to 0.300 for the event Propose/Request. The meaning in terms of probabilities or CDF's of increasing M from 0.1201 to 0.30 is not clear, but we only wish to model an "increased effort" in this state variable and increasing the value of M tends to convey this idea of "increased effort". The output from this 200 day simulation are shown in Figure 4.6

The results of this run demonstrate that there is no perceptible change in any of the system variables due to the increased effort. In fact, observe that even the state variable Propose/Request is unaffected. The value of M was subsequently increased several times, but the results were always similar to that displayed on Figure 4.6. We might conclude that the interactions between variables themselves are more important in the outcome of the simulation than are the initial starting points. Can we draw any relevant policy decision information from this simulation? Certainly one would not state, based on the results of this run, that an increased effort in Proposals has no benefit. However, the model does tend to indicate that the arms transfer process is quite resistant to change from simple event parameter changes.

3. **Extended Congressional Debate**

Another arms transfer policy proposed in chapter one is that the arms transfer process should be made more "viscous" by extending the length for Congressional debate. What might the effect on arms transfers be if this policy were implemented? The parameters for Evaluate/Negotiate are currently set at

$$t_h \text{ (time when CDF = 0.5)} = 55$$
$$t_e \text{ (time when CDF = 0.1)} = 40$$

To simulate extended Congressional debate these values can be modified to

$$t_h \text{ (time when CDF = 0.5)} = 75$$
$$t_e \text{ (time when CDF = 0.1)} = 35$$

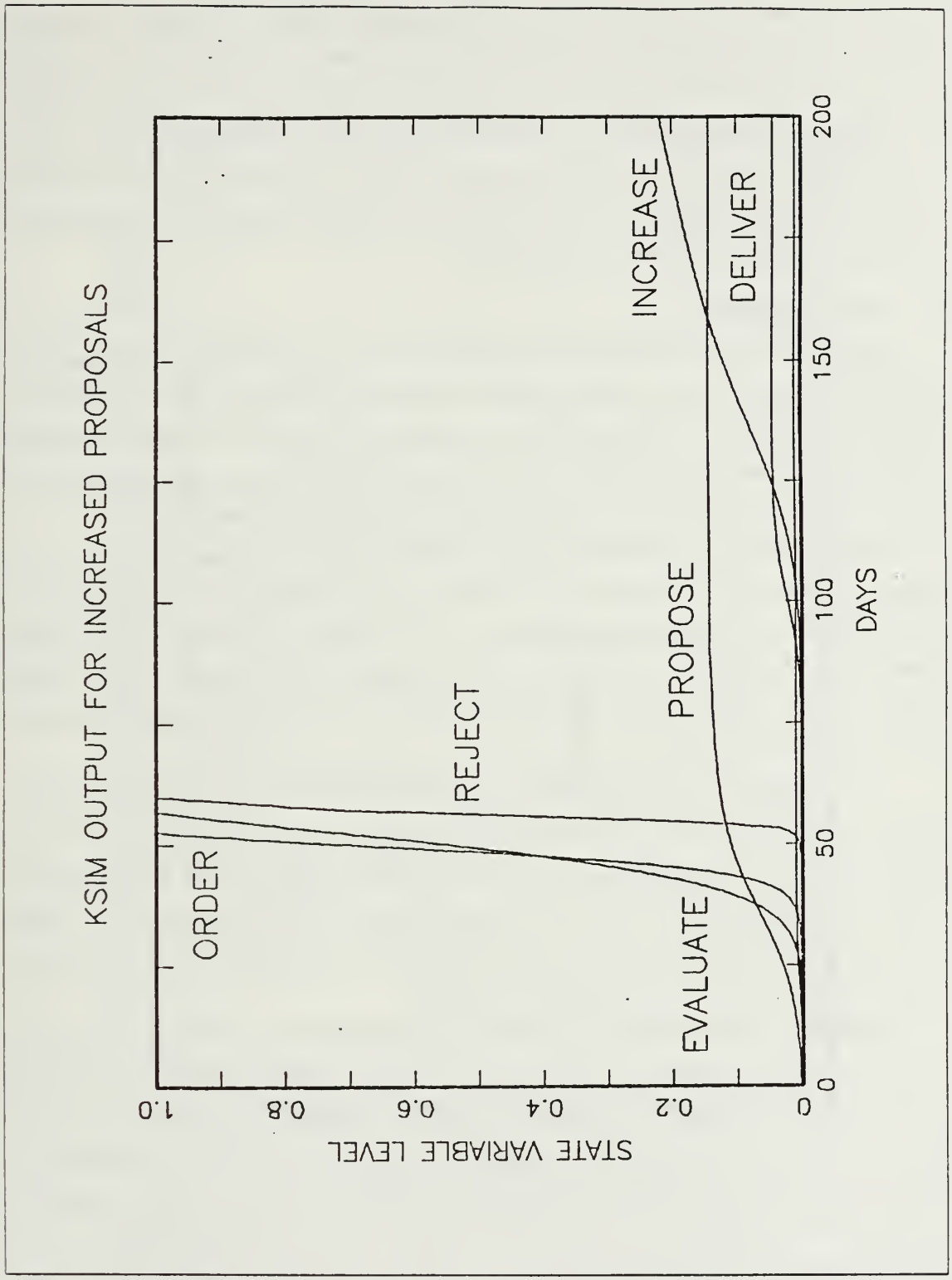

Figure 4.6 Results of Increased Effort in Propose/Request.

Again the model is run for 200 days. By examining Figure 4.7 it can be seen that extended Congressional debate does not have a great deal of effect on the arms transfer process. In fact, the only result other than making the growth of Evaluate/Negotiate less steep is to delay the growth of the state variable Reject/Refuse by about 5 days. We could argue again that the difference in model behavior from altering event parameters is so low that we cannot make any solid policy analysis recommendations based on the results of the simulation alone.

B. CONCLUSIONS

The KSIM model has demonstrated that it is not only feasible, but desireable to combine soft data, such as expert opinion, with hard data, such as the Third Point Systems event data set. This capability to augment hard data with subjective estimates is critical when modelling systems, like the arms transfer process, which are not well described by deterministic relationships with easily quantified parameters.

Another valuable implication of using KSIM is that it provides a method for the analyst to identify the structure of the system he is modelling, even though he may have no experience in modelling techniques. The process of estimating cross impact values and event parameters will give the user a great deal of insight into the "inner workings" of his system. Often this insight is just as valuable to the analyst as actual runs of the simulation itself.

The question of model validation is bound to arise when discussing forecasting models such as the KSIM simulation of arms transfers. The answer is simply that the whole purpose of the the model is to provide the foreign policy analyst with a means to develop a mental image of the arms transfer process. It is intended to be neither predictive nor prescriptive; merely a mathematically sound method for showing the geometric relationships between system variables.

While the KSIM procedure appears to be a useful tool in developing a model of the arms transfer process, the utility of this tool in modelling various arms transfer policies seems questionable. The model is very responsive to changes in the cross impact parameters, but resistant to changes in the event parameters. However, the various arms transfer policies are modelled by changing the relevant event parameters. Thus the policy analyst will not be presented with a clear picture of the results of his policy like those that were seen when cross impact parameters were changed. Since KSIM relies on the geometry of state variables, subtle changes in the values of the variables over time is not sufficient to draw conclusions about the effect of a given arms transfer policy.

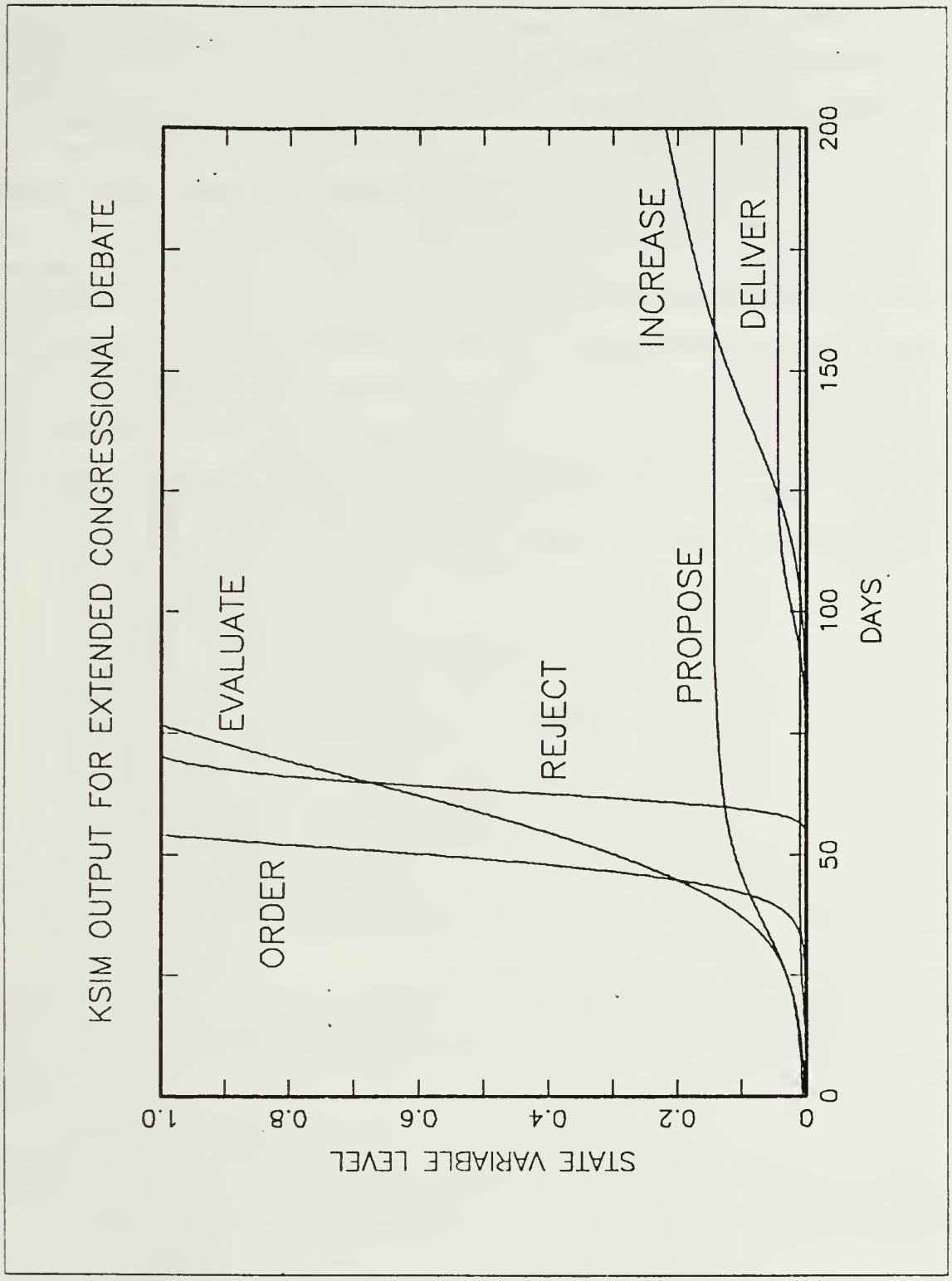

Figure 4.7 Results of Extended Congressional Debate.

Some comment must be made here regarding the implementation of a KSIM procedure with a panel of experts with limited mathematical experience. The ability to understand the difference between constant cross impacts and derivative cross impacts is very difficult to someone who does not possess a working knowledge of basic calculus. Thus, the policy analyst may easily become baffled when trying to estimate parameters for these cross impacts and when trying to understand why a state variable is so affected by the rate of growth of other variables . Furthermore, the process of arriving at a final set of cross impact parameters is one of trial and error. This is not only very time consuming, but the panel of experts have no knowledge of what the system behavior actually is; they only know what they think it should be or what they want it to be. Clearly, the final arms transfer model could be heavily biased and yield results that are completely inaccurate. To be really useful as an arms transfer policy analysis tool then, the KSIM procedure needs some method for obtaining the values of the cross impact terms from a hard data base.

APPENDIX
KSIM ARMS TRANSFER SIMULATION PROGRAM LISTING

Programed in Turbo Pascal (version 3.01a)

```
PROGRAM KSIM;
TYPE
      ParamRecord = RECORD
            THigh : REAL;
            Tlow : REAL;
            MaxProb : REAL;
      END;
VAR
      Done,SubMenu1,SubMenu2,SubMenu3: BOOLEAN;
      Event,Row,I,J,RndInitialT,T: INTEGER;
      StartTime,StopTime,MinT: INTEGER;
      BijValue,AijValue,Time,DeltaT: REAL;
      ThighVal,TlowVal,MaxProbVal: REAL;
      LowT,HighT,MaxP:REAL;
      Alpha,Beta,InitialT:REAL;
      Name: STRING[18];
      BijArray: ARRAY[1..8,1..8] of REAL;
      AijArray: ARRAY[1..8,1..8] of REAL;
      AlphaBeta: ARRAY[1..8,1..3] of REAL;
      Xvalue: ARRAY[1..8,1..2] OF REAL;
      PlotPoints: ARRAY[1..1000,1..8] of REAL;
      EventData: ARRAY[1..8,1..3] of REAL;
      Tpoint01: ARRAY [1..8] of REAL;
      EventNames: ARRAY[1..8] of STRING[18];
      ErrorTrap: ARRAY[1..8] of BOOLEAN;
      Underflow: ARRAY[1..8] of BOOLEAN;
      Overflow: ARRAY[1..8] of BOOLEAN;
      Params: ParamRecord;
      DataOut: FILE of ParamRecord;
      RawData: FILE of ParamRecord;
      BijFile,AijFile: TEXT;
      Names: TEXT;
      XvsTIME: TEXT;
      FILEN: TEXT;
FUNCTION POWER(Term,Exponent:REAL): REAL;   This function computes the value
BEGIN                                       of "term" raised to the power of
      POWER := EXP(Exponent*LN(Term));
END;
CONST
      Limit: INTEGER = 8;
      Empty: INTEGER = -1000;
PROCEDURE LoadEventNamesArray;                    Loads an array of the event
BEGIN                                             names for use in promting
      ASSIGN (Names, 'Events.Nam');                inputs and displaying some
      RESET (Names);                               of the results to the user.
      FOR Event := 1 TO Limit DO                   Events.Nam is on the disk.
            BEGIN
                  READLN(Names,Name);
                  EventNames[Event] := Name;
            END;
      CLOSE (Names);
END;
```

```
PROCEDURE LoadData;
 This procedure takes the estimates for times of highest probability
 and lowest probability (defined as that point where CDF = 0.10) and
 the maximum probability.  These values are stored on the disk in the
 file RawData.Dat.
BEGIN
      CLRSCR;
      GOTOXY(15,4);
      WRITELN('This procedure will ask you to estimate parameters');
      GOTOXY(15,5);
      WRITELN('about the events that go into an arms transfer.');
      ASSIGN (RawData, 'RawData.Dat');
      REWRITE (RawData);
      For Event := 1 to Limit DO
            BEGIN
                  GOTOXY(24,8);
                    WRITE ('For ',EventNames[Event], ' please enter:');
                  GOTOXY(15,12);
                    WRITE ('The time when the event is most likely to occur   '
                    READLN (ThighVal);
                    Params.Thigh := ThighVal;
                    EventData[Event,1] := ThighVal;
                  GOTOXY(15,14);
                    WRITE ('The time before which it probably will not occur
                    READLN (TlowVal);
                    Params.Tlow := TlowVal;
                    EventData[Event,2] := TlowVal;
                  GOTOXY(15,16);
                    WRITE ('The maximum probability of occurence   ');
                    READLN (MaxProbVal);
                    Params.MaxProb := MaxProbVal;
                    EventData[Event,3] := MaxProbVal;
                  WRITE (RawData, Params);
                  CLRSCR;
            END;
            CLOSE (RawData);
            WRITELN('Returning to Main Menu');
            DELAY(500);
 END;

PROCEDURE LoadBijMatrix;
 This procedure takes the user's estimates of cross impact values.  The Bij
 matrix is structured such that IMPACTING events are the columns and IMPACTED
 events are the rows.  Thus B[3,4] is the impact of event 4 on event 3.
 The matrix is loaded into an array, BijArray, and written to the disk.
BEGIN
      CLRSCR;
      GOTOXY(10,2);
      WRITELN('This procedure will load the B cross impact matrix.  For each of
      GOTOXY(10,3);
      WRITELN('the following event pairs, estimate the cross impact of the the'
      GOTOXY(10,4);
      WRITELN('first event on the second.  Positive values are enhancing, and')
      GOTOXY(10,5);
      WRITELN('negative values are inhibiting.  Use the following scale:');
      GOTOXY(33,7);WRITELN('0 ... NONE');
      GOTOXY(33,8);WRITELN('1 ... MILD');
      GOTOXY(33,9);WRITELN('2 ... STRONG');
      GOTOXY(33,10);WRITELN('3 ... OVERWHELMING');
      WRITELN;
      ASSIGN (BijFile,'BijData.Dat');
      REWRITE (BijFile);
      FOR Event := 1 to Limit DO
            BEGIN
                  FOR Row := 1 TO Limit DO
                        BEGIN
                              WRITE ('The impact of ',EventNames[Event],' on ',
                                    EventNames[Row],'   ');
```

```
                              READLN (BijValue);
                              BijArray[Row,Event] := BijValue;
                               See description of procedure for indexing of these
                               values
                              WRITE (BijFile,BijValue);
                              WRITELN;
                        END;
                  CLRSCR;
                  GOTOXY(25,2);WRITELN('0 ... NONE');
                  GOTOXY(25,3);WRITELN('1 ... MILD');          Writes the scale
                  GOTOXY(25,4);WRITELN('2 ... STRONG');        again so it is
                  GOTOXY(25,5);WRITELN('3 ... OVERWHELMING');  always available
                  WRITELN;
            END;
      CLRSCR;
      WRITELN ('Cross impact matrix is loaded');WRITELN;
      WRITELN ('Returning to main menu');
      DELAY (500);
END;

PROCEDURE LoadAijMatrix;
 This procedure takes the user's estimates of constant cross impact values.
 The Aij matrix is structured such that IMPACTING events are the columns
 columns and IMPACTED events are the rows.  Thus A[3,4] is the impact of event
 4 on event 3.  The matrix is loaded into an array, AijArray, and written
 to the disk.
BEGIN
      CLRSCR;
      GOTOXY(10,2);
      WRITELN('This procedure will load the A cross impact matrix.  For each of');
      GOTOXY(10,3);
      WRITELN('the following event pairs, estimate the cross impact of the the');
      GOTOXY(10,4);
      WRITELN('first event on the second.  Positive values are enhancing, and');
      GOTOXY(10,5);
      WRITELN('negative values are inhibiting.  Use the following scale:');
      GOTOXY(33,7);WRITELN('0.0 ... NONE');
      GOTOXY(33,8);WRITELN('0.1 ... MILD');
      GOTOXY(33,9);WRITELN('0.2 ... STRONG');
      GOTOXY(33,10);WRITELN('0.3 ... OVERWHELMING');
      WRITELN;
      ASSIGN (AijFile,'AijData.Dat');
      REWRITE (AijFile);
      FOR Event := 1 to Limit DO
            BEGIN
                  FOR Row := 1 TO Limit DO
                        BEGIN
                              WRITE ('The impact of ',EventNames[Event],' on ',
                                    EventNames[Row],'    ');
                              READLN (AijValue);
                              AijArray[Row,Event] := AijValue;
                               See description of procedure for indexing of these
                               values
                              WRITE (AijFile,AijValue);
                              WRITELN;
                        END;
                  CLRSCR;
                  GOTOXY(25,2);WRITELN('0 ... NONE');
                  GOTOXY(25,3);WRITELN('1 ... MILD');          Writes the scale
                  GOTOXY(25,4);WRITELN('2 ... STRONG');        again so it is
                  GOTOXY(25,5);WRITELN('3 ... OVERWHELMING');  always available
                  WRITELN;
            END;
      CLRSCR;
      WRITELN ('Cross impact matrix is loaded');WRITELN;
      WRITELN ('Returning to main menu');
      DELAY (500);
END;
```

```
PROCEDURE ComputeKSIMinputs;
 This procedure reads the raw data from RawData.Dat and computes Alpha, Beta
 and Tpoint01 (the time when the event CDF is at .01).  The results are in
 array AlphaBeta and Tpoint01.  AlphaBeta also holds the value of MaxProb.
CONST
      Log9: REAL = 2.19722457;
BEGIN
      MinT := 1000;  Arbitrarily large value so that the minimum Tpoint01 can
      Event := 0;                                        be found.
      CLRSCR;
      FOR Event := 1 to Limit DO
            BEGIN
                  HighT := Eventdata[Event,1];
                  LowT := Eventdata[Event,2];       Inputs from RawData.Da
                  MaxP := Eventdata[Event,3];
                  Alpha := 1/(HighT-LowT)*Log9;    Computation of paramete
                  Beta := HighT/(HighT-LowT)*Log9;
                  InitialT := (LN(1000*MaxP-1)-Beta)/(-Alpha);
                  If InitialT < MinT THEN MinT := ROUND(InitialT);
                  AlphaBeta[Event,1] := Alpha;
                  AlphaBeta[Event,2] := Beta;
                  AlphaBeta[Event,3] := MaxP;
                  Tpoint01[Event] := InitialT;
            END;
      CLOSE (RawData);
END;

PROCEDURE RetrieveBijMatrix;
 This procedure recalls the cross impact matrix from the disk
 when the system is intialized and builds the cross impact array
 (BijArray) for use in the simulation.
BEGIN
      ASSIGN (BijFile,'BijData.Dat');
      RESET (BijFile);
            FOR J := 1 to Limit DO
                  BEGIN
                        FOR I := 1 to Limit DO
                              BEGIN
                                    READ(BijFile,BijValue);
                                    BijArray[I,J] := BijValue;
                                     See description of the procedure LoadBijMa
                                     for the indexing of this array.
                              END;
                  END;
      CLOSE (BijFile);
END;

PROCEDURE RetrieveAijMatrix;
 This procedure recalls the cross impact matrix from the disk
 when the system is intialized and builds the cross impact array
 (BijArray) for use in the simulation.
BEGIN
      ASSIGN (AijFile,'AijData.Dat');
      RESET (AijFile);
            FOR J := 1 to Limit DO
                  BEGIN
                        FOR I := 1 to Limit DO
                              BEGIN
                                    READ(AijFile,AijValue);
                                    AijArray[I,J] := AijValue;
                                     See description of the procedure LoadAijMa
                                     for the indexing of this array.
                              END;
                  END;
      CLOSE (AijFile);
END;

PROCEDURE RetrieveEventData;
```

```
 This procedure recalls the raw data from the disk when the program
 is intialized.  KSIM inputs (alpha, beta, and initial time) are
 computed immediately by procedure ComputeKISMinputs.
BEGIN
      ASSIGN (RawData, 'RawData.Dat');
      RESET (RawData);
      FOR Event := 1 to Limit DO
            BEGIN
                  READ (RawData,Params);
                  WITH Params DO
                        EventData[Event,1] := Params.THigh;
                        EventData[Event,2] := Params.TLow;
                        EventData[Event,3] := Params.MaxProb;
            END;
      CLOSE (RawData);
END;

PROCEDURE DisplayEventData;
 This procedure displays the current values of event data on the screen.
CONST
      C: INTEGER = 6;
VAR
      Y: INTEGER;
      Return: CHAR;
BEGIN
      CLRSCR;
      GOTOXY(25,2);WRITELN('E V E N T   P A R A M E T E R S');
      GOTOXY(25,4);WRITELN('Most       Not');                        Column
      GOTOXY(24,5);WRITELN('Likely     Before       Maximum');       headers
      GOTOXY(25,6);WRITELN('Time       Time      Probability');
      FOR Event := 1 to Limit DO
            BEGIN
                  Y := 2*Event+C;
                  GOTOXY(2,Y);WRITELN(EventNames[Event]);   Row headers
                  GOTOXY(25,Y);WRITELN(EventData[Event,1]:3:0,'         ',
                                       EventData[Event,2]:3:0,'          ',   Current
                                       EventData[Event,3]:4:4);              values
            END;
      GOTOXY(20,24);WRITE('Hit "ENTER" to return to main menu');
      READLN (Return);
END;

PROCEDURE DisplayBijMatrix;
 This procedure displays the current cross impact matrix on the screen.
CONST
      C1: INTEGER = 6;
      C2: INTEGER = 16;
VAR
      X,Y: INTEGER;
      Return: CHAR;
BEGIN
      CLRSCR;
      GOTOXY(23,2);WRITELN('C R O S S    I M P A C T   M A T R I X');
      GOTOXY(36,4);WRITELN('IMPACTING EVENTS');    Column headers
      GOTOXY(23,6);WRITELN('1     2     3     4     5     6     7     8');
      FOR I := 1 to Limit DO
            BEGIN
                  Y := 2*I+C1;
                  IF I=4 THEN
                        BEGIN
                              GOTOXY(5,Y);WRITELN('IMPACTED');    Row header
                        END;
                  IF I=5 THEN
                        BEGIN
                              GOTOXY(6,Y);WRITELN('EVENTS');      Row header
                        END;
                  GOTOXY(18,Y);WRITELN(I);
```

```
                FOR J := 1 to Limit DO
                        BEGIN
                                X := 6*J+C2;
                                GOTOXY(X,Y);WRITELN(BijArray[I,J]:2:1);   Current
                        END;                                                values
        END;
    GOTOXY(28,24);WRITE('Hit "ENTER" to return to main menu');
    READLN (Return);
END;

PROCEDURE DisplayAijMatrix;
 This procedure displays the current cross impact matrix on the screen.
CONST
    C1: INTEGER = 6;
    C2: INTEGER = 16;

VAR
    X,Y: INTEGER;
    Return: CHAR;

BEGIN
    CLRSCR;
    GOTOXY(23,2);WRITELN('C R O S S    I M P A C T   M A T R I X');
    GOTOXY(36,4);WRITELN('IMPACTING EVENTS');    Column headers
    GOTOXY(23,6);WRITELN('1     2     3     4     5     6     7     8');
    FOR I := 1 to Limit DO
        BEGIN
            Y := 2*I+C1;
            IF I=4 THEN
                BEGIN
                    GOTOXY(5,Y);WRITELN('IMPACTED');    Row header
                END;
            IF I=5 THEN
                BEGIN
                    GOTOXY(6,Y);WRITELN('EVENTS');      Row header
                END;
            GOTOXY(18,Y);WRITELN(I);
            FOR J := 1 to Limit DO
                BEGIN
                    X := 6*J+C2;
                    GOTOXY(X,Y);WRITELN(AijArray[I,J]:2:1);   Current
                END;                                            values
        END;
    GOTOXY(28,24);WRITE('Hit "ENTER" to return to main menu');
    READLN (Return);
END;

PROCEDURE EditEventData;
 This procedure allows the user to change current values of the event data.
 The current values are displayed and new values are input by the user.  The
 new value then replaces the old value in the event data array.
CONST
    C: INTEGER = 6;

VAR
    Y: INTEGER;
    NewValue: REAL;

BEGIN
    CLRSCR;
    GOTOXY(21,3);WRITELN('E V E N T   D A T A   E D I T O R');
    FOR Event := 1 TO LIMIT DO
        BEGIN
            Y := 2*Event+C;
            GOTOXY(28,Y);WRITELN(Event,'.  ',EventNames[Event]);
        END;
  GOTOXY(16,(Y+3));WRITE('Enter the number of the event you want changed.   ');
    READLN(Event);
    CLRSCR;
    GOTOXY(22,8);WRITELN('Current values for ',Eventnames[Event]);
  GOTOXY(22,11);WRITELN('1.  Most likely time    = ',EventData[Event,1]:3:0);
```

```
   GOTOXY(22,13);WRITELN('2.  Not Before time     = ',EventData[Event,2]:3:0);
   GOTOXY(22,15);WRITELN('3.  Maximum probability = ',EventData[Event,3]:4:4);
       GOTOXY(22,19);WRITE('Enter the number you want to change.   ');
       READLN (I);
       GOTOXY(22,21);WRITE('Enter the new value.    ');
       READLN (NewValue);
       EventData[Event,I] := NewValue;
       GOTOXY(22,23);WRITELN('Change complete ... returning to main menu');
       DELAY(1500);
END;

PROCEDURE EditBijMatrix;
 This procedure allows the user to change current values of the cross
 impact matrix.  First the impacting event is selected, then the impacted
 event is selected.  The new value is input and repaces the old value in
 the BijMatrix array.
CONST
       C: INTEGER = 6;

VAR
       Y: INTEGER;
       NewValue: REAL;

BEGIN
       CLRSCR;
       GOTOXY(15,3);
       WRITELN('C R O S S   I M P A C T   M A T R I X   E D I T O R');
       FOR Event :=1 to Limit DO
             BEGIN
                Y := 2*Event+C;
                GOTOXY(30,y);WRITELN(Event,'.  ',EventNames[Event]);
             END;
       GOTOXY(20,(Y+3));WRITE('Enter the number of the IMPACTING event.   ');
       READLN (J);
       CLRSCR;
       FOR Event :=1 to Limit DO
             BEGIN
                Y := 2*Event+C;
                GOTOXY(30,Y);WRITELN(Event,'.  ',EventNames[Event]);
             END;
       GOTOXY(20,(Y+3));WRITE('Enter the number of the IMPACTED event.   ');
       READLN (I);
       CLRSCR;
       GOTOXY(20,13);WRITE('Enter the new value for the impact of');
       GOTOXY(20,15);WRITE(EventNames[J],'   on   ',EventNames[I],'   ');
       READLN (NewValue);
       BijArray[I,J] := NewValue;
       GOTOXY(20,18);WRITELN('Change complete ... returning to main menu');
       DELAY (1500);
END;

PROCEDURE EditAijMatrix;
 This procedure allows the user to change current values of the cross
 impact matrix.  First the impacting event is selected, then the impacted
 event is selected.  The new value is input and repaces the old value in
 the BijMatrix array.
CONST
       C: INTEGER = 6;

VAR
       Y: INTEGER;
       NewValue: REAL;

BEGIN
       CLRSCR;
       GOTOXY(15,3);
       WRITELN('C R O S S   I M P A C T   M A T R I X   E D I T O R');
       FOR Event :=1 to Limit DO
             BEGIN
                Y := 2*Event+C;
                GOTOXY(30,y);WRITELN(Event,'.  ',EventNames[Event]);
```

```
            END;
      GOTOXY(20,(Y+3));WRITE('Enter the number of the IMPACTING event.   ');
      READLN (J);
      CLRSCR;
      FOR Event :=1 to Limit DO
            BEGIN
               Y := 2*Event+C;
               GOTOXY(30,Y);WRITELN(Event,'.  ',EventNames[Event]);
            END;
      GOTOXY(20,(Y+3));WRITE('Enter the number of the IMPACTED event.   ');
      READLN (I);
      CLRSCR;
      GOTOXY(20,13);WRITE('Enter the new value for the impact of');
      GOTOXY(20,15);WRITE(EventNames[J],'   on   ',EventNames[I],'   ');
      READLN (NewValue);
      AijArray[I,J] := NewValue;
      GOTOXY(20,18);WRITELN('Change complete ... returning to main menu');
      DELAY (1500);
END;

PROCEDURE InitializePlotPointsArray;
 The array PlotPoints holds the values of each event at each time increment.
BEGIN
      FOR T := 0 to 1000 do
            BEGIN
                  FOR Event := 1 to Limit DO
                       PlotPoints[T,Event] := 0;
            END;
END;

PROCEDURE InitializeTime;
 This procedure finds the start time for the simulation.  It selects the
 minimum Tpoint01 unless that value is less than 0.  If this occurs then
 the start time is set at 0 and CDF values are computed for those events
 with Tpoint01 less than 0.
BEGIN
      InitializePlotPointsArray;
      FOR Event := 1 to Limit DO
            BEGIN
                  ErrorTrap[Event] := FALSE;
                  Underflow[Event] := FALSE;
                  Overflow[Event] := FALSE;
                  Xvalue[Event,1] := 0;   Initializes Xvalue array.  Xvalue[t,1]
                  Xvalue[Event,2] := 0;   is X(t + dt).  Xvalue[t,2] is X(t).
            END;
      CLRSCR;
      GOTOXY(17,12);WRITE('How many days do you want to simulate? (250 max)  ');
      READLN(StopTime);
      DeltaT := 0.25;
      FOR Event := 1 to Limit DO
            BEGIN
                  Alpha := AlphaBeta[Event,1];
                  Beta := AlphaBeta[Event,2];
                  MaxP := AlphaBeta[Event,3];
                  IF Tpoint01[Event] < 0 THEN
                        BEGIN
                              Tpoint01[Event] := Empty;
                              Xvalue[Event,1] := MaxP/(1 + EXP(Beta));  CDF at t
                              Xvalue[Event,2] := MaxP/(1 + EXP(Beta));  CDF at t
                               Computes X(0) since T(.01) < 0
                        END;
            END;
            IF MinT < 0 THEN StartTime := 0 ELSE StartTime := MinT;
            GOTOXY(30,16);WRITELN('Start time will be ',StartTime);
END;


PROCEDURE IterativeEquation;
```

```
 This procedure computes the value of X(t + dt) based on the value of X(t).
 It takes into account cross impacts.  The equations are described in CHAPTER
 3 of the thesis.
VAR
      GammaSumPlus,GammaSumMinus : REAL;
      Gamma,PIsubI,dXdt,Denom,Num : REAL;
      XTplusDeltaT : REAL;
BEGIN
      Alpha := AlphaBeta[I,1];
      Beta := AlphaBeta[I,2];
      GammaSumPlus := 0;
      GammaSumMinus := 0;
      FOR J:=1 to Limit DO
            BEGIN
                  IF (Xvalue[J,1] = Xvalue[J,2]) THEN Gamma:=0
                  ELSE
                        BEGIN
                              dXdt := (Xvalue[J,1]-Xvalue[J,2])/DeltaT;
                               dXdt = X(t + dt) - X(t) / dt
                              Gamma := AijArray[I,J]*Xvalue[J,1]+BijArray[I,J]*dXdt;
                        END;
                  GammaSumMinus := GammaSumMinus + ABS(Gamma) - Gamma;
                  GammaSumPlus := GammaSumPlus + ABS(Gamma) + Gamma;
            END;
     PisubI := (1 + 0.5*DeltaT*GammaSumMinus)/(1 + 0.5*DeltaT*GammaSumPlus);
     IF ErrorTrap[I] THEN   prevents underflow error with EXP ni Num and Denom
           BEGIN
                 Num := 0.0;
                 Denom := 1.0;
           END
           ELSE
                 BEGIN
                       Num := Alpha*Deltat*EXP(-Alpha*Time + Beta);
                       Denom := 1 + EXP(-Alpha*Time + Beta);
                       IF (EXP(-Alpha*Time + Beta) < 1E-20) THEN   Underflow error
                             ErrorTrap[I] := TRUE;                  is immenent.
                 END;
     IF Underflow[I] THEN XTplusDeltaT := 0.0
           ELSE
                 IF Overflow[I] THEN XTplusDeltaT := 1.0
                       ELSE
                 BEGIN
     XTplusDeltaT := POWER(Xvalue[I,1],PIsubI)*POWER((1 + Num/Denom),PIsubI);
     IF (Xvalue[I,1] < 1E-20) THEN Underflow[I] := TRUE;
     IF (Xvalue[I,1] > 0.999999) THEN Overflow[I] := TRUE;
                 END;
     Xvalue[I,2] := Xvalue[I,1];   Replaces X(t) with old X(t + dt)
     Xvalue[I,1] := XTplusDeltaT;  New X(t + dt) for next time increment
     PlotPoints[T,I] := Xvalue[I,2];
END;

PROCEDURE Simulation;
 The simulation sets X(t) = 0 if the time is before T(.01) for X.  When time
 is equal to or greater to T(.001) then X(t) is set to 0.001.  Thereafter, the
 procedure IterativeEquation computes values for X(t + dt).
CONST
      Zero: REAL = 0.0;
BEGIN
      Time := StartTime;
      T := 1;
      WHILE Time <= StopTime DO
      BEGIN
            GOTOXY(35,18);WRITELN('Time = ',Time:3:1);
            For I:=1 to Limit DO
                  IF (Tpoint01[I]<=Time) AND (Tpoint01[I]<>Empty) THEN
                   Checks if T(.01) is less than present time.  If this value
                   is less than present time OR if it is empty, these steps are
                   skipped.
```

```
                        BEGIN
                              Xvalue[I,1] := 0.001; Both are set at 0.001 so swi
                              Xvalue[I,2] := 0.001; in IterativeEquation will wo
                              Tpoint01[I] := Empty; Sets EMPTY value so above ch
                        END;                        will be skipped.
            For I:=1 to Limit DO
                  BEGIN
                        IF Xvalue[I,1] = Zero THEN
                        BEGIN
                              CASE I of
                                    1: Plotpoints[T,1] := 0;
                                    2: Plotpoints[T,2] := 0;
                                    3: Plotpoints[T,3] := 0;
                                    4: Plotpoints[T,4] := 0;
                                    5: Plotpoints[T,5] := 0;
                                    6: Plotpoints[T,6] := 0;
                                    7: Plotpoints[T,7] := 0;
                                    8: Plotpoints[T,8] := 0;
                              END;
                        END
                        ELSE IterativeEquation;

                  END;
                  Time := Time + DeltaT;    Increments time
                  T := T + 1                Increments PlotPoints array row inde
      END;
END;

PROCEDURE PrintResults;
 Sends results to the printer
BEGIN
      CLRSCR;
      GOTOXY(25,15);WRITELN('Sending results to printer.');
      DELAY(500);
      WRITE(LST,'Time       1        2        3        4       ');
      WRITELN(LST,'5        6        7        8');
      Time := StartTime;
      T := 1;
      WHILE Time <= StopTime DO
            BEGIN
                  WRITE (LST,Time:4:1,'    ');
                  WRITE (LST,PlotPoints[T,1]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,2]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,3]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,4]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,5]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,6]:4:4,'    ');
                  WRITE (LST,PlotPoints[T,7]:4:4,'    ');
                  WRITELN (LST,PlotPoints[T,8]:4:4);
                  Time := Time + 8*DeltaT;
                  T := T + 8;
            END;
            GOTOXY(30,17);
            WRITELN('Enter To Continue');
            READLN;
END;

PROCEDURE WriteOutputFile;
 This procedure writes a sequential data file to the disk.  The user inputs
 the name of the file.  The procedure writes time, X1 values, X2 values, etc.
 Time is multiplied by 100 and rounded, X values are multiplied by 10000 and
 rounded.  This is so that the data is not in exponential notation.  The data
 can then be plotted on high resolution plotters such as GRAFSTAT.
VAR
      Column,RndPoint,RndTime: INTEGER;
      Point: REAL;
      OutFileName: STRING[14];
BEGIN
      CLRSCR;
```

```
      GOTOXY(25,14);WRITE ('What name do you want for the output file?   ');
      GOTOXY(25,17);
      READLN (OutFileName);
      ASSIGN (XvsTIME, OutFileName);
      REWRITE (XvsTIME);
      Time := StartTime;
      REPEAT
            RndTime := ROUND(Time*100);
            WRITELN(XvsTIME, RndTime);
            Time := Time + DeltaT;
      UNTIL Time > StopTime;
      FOR Column := 1 to Limit DO
            BEGIN
                  Time := StartTime;
                  T := 1;
                  REPEAT
                        Point := PlotPoints[T,Column];
                        RndPoint := ROUND(Point*10000);
                        WRITELN (XvsTIME, RndPoint);
                        Time := Time + DeltaT;
                        T := T + 1;
                  UNTIL Time > StopTime;
            END;
      CLOSE (XvsTIME);
END;
PROCEDURE DisplayMainMenu;
BEGIN
      SubMenu1 := FALSE;  Preset to FALSE. Selection of a menu choice
      SubMenu2 := FALSE;  1,2 or 3 will switch and select proper submenu
      SubMenu3 := FALSE;  interpretation.
      CLRSCR;
      GOTOXY(20,5);WRITELN ('CROSS - IMPACT ANALYSIS FOR ARMS TRANSFERS');
      GOTOXY(35,8);WRITELN ('MAIN MENU');
      GOTOXY(28,10);WRITELN ('1.  Enter New Data');
      GOTOXY(28,12);WRITELN ('2.  Review Current Data');
      GOTOXY(28,14);WRITELN ('3.  Edit Current Data');
      GOTOXY(28,16);WRITELN ('4.  Run Model');
      GOTOXY(28,18);WRITELN ('5.  Write Output File for GRAFSTAT');
      GOTOXY(28,20);WRITELN ('6.  Exit Program');
      GOTOXY(20,23);WRITE ('Type number would you like, then hit ENTER   ');
END;

PROCEDURE DisplaySubMenu;
BEGIN
      CLRSCR;
      GOTOXY(32,8);WRITELN ('SUB MENU');
      GOTOXY(25,10);WRITELN ('1.  Event Parameters');
      GOTOXY(25,12);WRITELN ('2.  Constant Cross Impact Matrix');
      GOTOXY(25,14);WRITELN ('3.  Derivative Cross Impact Matrix');
      GOTOXY(25,16);WRITELN ('4.  Return to Main Menu');
      GOTOXY(17,21);WRITE ('Type number would you like, then hit ENTER   ');
END;

PROCEDURE ExitProgram;
 This program writes the current values of the cross impact matrix and
 the event data to the disk and exits the program.
BEGIN
      CLRSCR;
      ASSIGN (BijFile,'BijData.Dat');
      REWRITE (BijFile);
            FOR J := 1 TO Limit DO
                  BEGIN
                        FOR I := 1 TO Limit DO
                              BEGIN
                                    BijValue := BijArray[I,J];
                                    WRITE (BijFile,BijArray[I,J]);
                              END;
                  END;
```

```
      CLOSE (BijFile);
      ASSIGN (AijFile,'AijData.Dat');
      REWRITE (AijFile);
            FOR J := 1 TO Limit DO
                  BEGIN
                        FOR I := 1 TO Limit DO
                              BEGIN
                                    AijValue := AijArray[I,J];
                                    WRITE (AijFile,AijArray[I,J]);
                              END;
                  END;
      CLOSE (AijFile);
      ASSIGN (RawData,'RawData.Dat');
      REWRITE (RawData);
      FOR I := 1 TO Limit DO
            BEGIN
                  Params.THigh := EventData[I,1];
                  Params.TLow := EventData[I,2];
                  Params.MaxProb := EventData[I,3];
                  WRITE (RawData,Params);
            END;
      CLOSE (RawData);
      GOTOXY(30,14);WRITELN('END OF ANALYSIS');
      DELAY(1500);
      CLRSCR;
      Done := TRUE;
END;

PROCEDURE ErrorFlag;
 This procedure is activated if the user selects an invalid choice from
 the menu.  It always returns to the main menu.
BEGIN
      WRITELN ('INVALID MENU SELECTION ... CHOOSE AGAIN');
      DELAY (700);  Allows user to read message.
END;

PROCEDURE InterpretMainOption;
 Interprets menu selection from the main menu
VAR
      Option : INTEGER;
      Response : CHAR;
BEGIN
      READLN (Option);
      CASE Option OF
            1: BEGIN
                  SubMenu1 := TRUE;   Insures proper sub menu actions
                  DisplaySubMenu;
                  END;
            2: BEGIN
                  SubMenu2 := TRUE;
                  DisplaySubMenu;
                  END;
            3: BEGIN
                   SubMenu3 := TRUE;
                   DisplaySubMenu;
                   END;
            4: BEGIN
                   ComputeKSIMinputs;  Computes event parameters
                   InitializeTime;
                   Simulation;         Actual KSIM model in here
                   GOTOXY(28,20);WRITE('Print Results? y for YES    ');
                   READLN(Response);
                   IF Response = 'y' THEN PrintResults;
               END;
            5: WriteOutputFile;
            6: ExitProgram;
      ELSE ErrorFlag;   Returns to the main menu
      END;
END;
```

```
PROCEDURE InterpretOption1;
 This procedure activated when #1 is the main menu choice.  These
 perform the sub menu actions requested.
VAR
      SubOption : INTEGER;
BEGIN
      READLN (SubOption);
      CASE SubOption OF
            1:  BEGIN
                        LoadData;
                        ComputeKSIMinputs;
                 END;
            2:  LoadAijMatrix;
            3:  LoadBijMatrix;
            4:  BEGIN
                        CLRSCR;
                        GOTOXY(30,15);WRITELN('Returning to Main Menu');
                        DELAY (500);
                 END;
      ELSE ErrorFlag;  Returns to the main menu
      END;
END;

PROCEDURE InterpretOption2;
 This procedure activated when #2 is the main menu choice.  These
 display the data requested.
VAR
      SubOption : INTEGER;
BEGIN
      READLN (SubOption);
      CASE SubOption OF
            1:  DisplayEventData;
            2:  DisplayAijMatrix;
            3:  DisplayBijMatrix;
            4:  BEGIN
                        CLRSCR;
                        GOTOXY(30,15);WRITELN('Returning to Main Menu');
                        DELAY (800);
                 END;
      ELSE ErrorFlag;  Returns to the main menu
      END;
END;

PROCEDURE InterpretOption3;
 This procedure activated when #3 is the main menu choice.  These
 perform editing on the data base requested.
VAR
      SubOption : INTEGER;
BEGIN
      READLN (SubOption);
      CASE SubOption OF
            1:  EditEventData;
            2:  EditAijMatrix;
            3:  EditBijMatrix;
            4:  BEGIN
                        CLRSCR;
                        GOTOXY(30,15);WRITELN('Returning to Main Menu');
                        DELAY (800);
                 END;
      ELSE ErrorFlag;  Returns to the main menu
      END;
END;

 MAIN PROGRAM
BEGIN
         TEXTBACKGROUND (1);
         TEXTCOLOR (14);
```

```
Done := False;
RetrieveEventData;        Initializes the
RetrieveBijMatrix;        system with the
RetrieveAijMatrix;        current values.
LoadEventNamesArray;
WHILE NOT Done Do
BEGIN
     DisplayMainMenu;
     InterpretMainOption;
     IF SubMenu1 THEN InterpretOption1;
     IF Submenu2 THEN Interpretoption2;
     IF Submenu3 THEN InterpretOption3;
END;
```

LIST OF REFERENCES

1. Hessing, Anne and others, *Controlling Future Arms Trade*, McGraw-Hill Book Company, New York, 1977.

2. Pierre, Andrew, *The Global Politics of Arms Sales*, Princeton University Press, Princeton, New Jersey, 1982.

3. Louscher, David, Salomone, Michael, *Marketing Security Assistance*, Lexington Books, 1987.

4. Enzer, Selwyn, "Cross-Impact Techniques in Technology Assessment," *Futures*, v. 4, pp. 30-51, March 1972.

5. Helmer, Olaf, "Problems in Futures Research: Delphi and Causal Cross-Impact Analysis," *Futures*, v. 9, pp. 17-31.

6. Martino, Joseph P., *Technological Forecasting for Decision Making*, American Elsevier Publishing Company, Inc., 1972.

7. Martino, Joseph P. and Chen, Kuei-Lin, "Cluster Analysis of Cross-Impact Model Scenarios," *Technological Forecasting and Social Change*, v. 12, pp. 61-71, 1978.

8. Gordon, T. J. and Hayward, H., "Initial Experiments with the Cross Impact Method of Forecasting," *Futures*, v. 1, pp. 100-116, December 1968.

9. Turoff, Murray, "An Alternate Approach to Cross Impact Analysis," *Technological Forecasting and Social Change* v. 3, pp. 309-339, 1972.

10. Alter, Steven, "The Evaluation of Generic Cross-Impact Models," *Futures*, v. 11, pp. 132-150, April 1979.

11. Bloom, Mitchel, "Deterministic Trend Cross-Impact Forecasting," *Technological Forecasting and Social Change*, v. 8, pp. 35-74, 1975.

12. Bloom, Mitchel, "Time Dependant Event Cross-Impact Analysis: Results from a New Model," *Technological Forecasting and Social Change*, v. 10, pp. 181-201, 1977.

13. Duperrin, J. C., and Godet, M., "SMIC-74 - A Method for Constructing and Ranking Scenarios," *Futures*, v. 7, pp. 302-312, August 1975.

14. Enzer, Selwyn and Alter, Steven. "Cross-Impact Analysis and Classical Probability: the question of consistency." *Futures*, v. 10, pp. 227-238, June 1978.

15. Duval, A., Fontela, E. and Gabus, A. "Cross-Impact Analysis: A Handbook on Concepts and Applications." in *Portraits of Complexity*: *Applications of Systems Methodologies to Societal Problems*. Maynard M. Baldwin (.ed), Battelle Memorial Institute, June 1975.

16. Mitchell, R. B. and Tydeman, J., "A Note on SMIC-74," *Futures*, v. 8, pp. 64-67, February 1976.

17. Kaya, Y., and others, "A Revised Cross-Impact Method and its Applications to the Forecast of Urban Transportation Technology," *Technological Forecasting and Social Change*, v. 14, pp.243-257, 1979.

18. Mitchell, R. B. and Tydeman, J., "Subjective Conditional Probability Modelling," *Technological Forecasting and Social Change*, v. 11, pp. 133-152, 1978.

19. Kane, Julius, "A Primer for a New Cross-Impact Language - KSIM," *Technological Forecasting and Social Change*, v. 4, pp. 129-142, 1972.

20. Lipinski, H. and Tydeman, M., "Cross Impact Analysis: Extended KSIM," *Futures*, v. 11, pp. 151-154, April 1979.

21. Kane, J., Vertinsky, I. and Thomson, W., "KSIM: A Methodology for Interactive Resource Policy Simulation.", *Water Resources Research*, v. 9, n. 1, pp. 65-79, February 1973.

22. Kane, J., Thompson, W., Vertinsky, I., "Health Care Delivery: A Policy Simulator," *Socio-Economic Planning Sciences*, v. 6, pp. 283-293, 1972.

23. Mather, John, "Simulation for Policy-Making," *Toward Real Time Simulation Languages, Models and Systems*, Part I, pp. 15-21, May 1975.

24. *Arms Transfer Handbook*, Version 6.0, Third Point Systems, Inc., Monterey, California, January 19, 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	F. Russell Richards Evaluation Technology Incorporated 2150 Garden Rd., Ste. B3 Monterey, CA 93940-5327	2
4.	LT Michael R. Mara 3506 Beverly Dr. Annandale, VA 22003	1

Thesis
M33545 Mara
c.1 Modelling arms transfers: an application of KSIM cross impact analysis.

Thesis
M33545 Mara
c.1 Modelling arms transfers: an application of KSIM cross impact analysis.

AMERICAN BINDERY
TOPEKA KANSAS

thesM33545
Modelling arms transfers:
3 2768 000 77498 8
DUDLEY KNOX LIBRARY